

# ***AWR2544 Technical Reference Manual***

## ***Technical Reference Manual***

---



Literature Number: SWRU611A  
JANUARY 2024 – REVISED MAY 2025





# Table of Contents



<b>1 Introduction</b>	13
1.1 Device Overview	13
1.2 Device Block Diagram	15
<b>2 Memory Map</b>	17
2.1 Main Subsystem Cortex R5F Memory Map	17
2.2 Radar Subsystem Memory Map	19
<b>3 System Interconnect</b>	22
3.1 Main Subsystem R5F Infrastructure	22
3.2 DSP Subsystem C66x Infrastructure	24
3.3 Radar Subsystem Infrastructure	27
3.4 Peripheral Central Resource (PCR)	27
3.4.1 MSS_PCR1 Registers	31
3.4.2 MSS_PCR2 Registers	343
3.4.3 DSS_PCR Registers	655
<b>4 Device Initialization</b>	967
4.2 Boot Process	967
4.2.1 ROM Code Overview	967
4.2.2 Boot Modes	967
4.2.3 SOP Mode Pins	968
4.2.4 BOOT-ROM Architecture (RBL)	968
4.2.5 R5F SBL Loading	969
<b>5 Device Configuration</b>	971
5.1 Overview	971
5.2 Control Registers	971
5.2.1 MSS_TOPRCM Registers	972
5.2.2 MSS_RCM Registers	1146
5.2.3 DSS_RCM Registers	1301
5.2.4 RSS_RCM Registers	1349
5.2.5 DSS_CTRL Registers	1387
5.2.6 RSS_CTRL Registers	1456
5.2.7 MSS_CTRL Registers	1698
5.2.8 MSS_IOMUX Registers	1988
5.2.9 TOP_CTRL Registers	2116
5.2.10 RSS_PROC_CTRL Registers	2229
5.2.11 DSP_ICFG Registers	2313
5.3 Device Clock Architecture	2481
5.3.1 Clock Overview	2481
5.3.2 Root Clocks	2483
5.3.3 External Clocks	2484
5.3.4 Clock Selection	2485
5.3.5 Analog Modules	2492
5.4 HS Divider Programming Sequence	2494
5.5 M2 and N2 Change On-the-Fly	2494
5.6 Clock Register Details	2494
5.7 Ethernet Clock	2495
5.8 Memory Protection Unit (MPU) and Firewalls	2496
5.8.1 Functional Operation	2496
5.8.2 Firewalls	2497
5.8.3 ID Allocation	2499

5.8.4 Memory Map (MPU).....	2500
5.9 Power Domains.....	2504
5.9.1 DSP Subsystem L3 Power Domain.....	2505
5.9.2 MSS L2 Power Domain.....	2505
5.10 Resets.....	2507
5.11 Temperature Sensor.....	2509
<b>6 Processors and Accelerators.....</b>	<b>2510</b>
6.1 Main Subsystem Cortex R5F.....	2510
6.1.1 Main ARM SubSystem Features.....	2510
6.1.2 TCM Initialization.....	2511
6.1.3 Resets.....	2511
6.1.4 MSS_CCMR Registers.....	2513
6.2 Radar Processing Hardware Accelerator.....	2520
6.2.1 Radar Accelerator Features Overview.....	2520
6.2.2 Radar Accelerator Architecture.....	2521
6.3 Radar Subsystem.....	2521
6.3.1 BIST Subsystem.....	2521
6.3.2 Clock Subsystem.....	2526
6.3.3 Transmit Subsystem.....	2527
6.3.4 Receive Subsystem.....	2528
6.3.5 Real(I-only) Receiver in AWR2544 .....	2528
<b>7 Interrupts.....</b>	<b>2530</b>
7.1 Main Subsystem Cortex R5F Interrupt Map.....	2530
7.2 Main Subsystem ESM Interrupt Map.....	2535
<b>8 ADC Buffer.....</b>	<b>2542</b>
8.1 Functional Description.....	2542
8.1.1 DFE Data Write Operation.....	2542
8.1.2 Test Pattern Generator Support.....	2544
8.1.3 ADC Buffer Data Formats.....	2544
<b>9 Ethernet SubSystem (MCU_CPSW0).....</b>	<b>2545</b>
9.1 MCU_CPSW0 Overview.....	2546
9.1.1 MCU_CPSW0 Features.....	2546
9.1.2 MCU_CPSW0 Not Supported Features.....	2547
9.1.3 Terminology.....	2547
9.2 MCU_CPSW0 Environment.....	2549
9.2.1 MCU_CPSW0 RMII Interface.....	2549
9.2.2 MCU_CPSW0 RGMII Interface.....	2550
9.3 MCU_CPSW0 Integration.....	2552
9.4 MCU_CPSW0 Functional Description.....	2554
9.4.1 Functional Block Diagram.....	2554
9.4.2 CPSW Ports.....	2554
9.4.3 Clocking.....	2555
9.4.4 Interrupt Functionality.....	2556
9.4.5 Media Independent Interfaces (XGMII/GMII).....	2558
9.4.6 Software IDLE.....	2560
9.4.7 CPSW_2G.....	2561
9.4.8 CPPI Checksum Offload.....	2582
9.4.9 Jumbo packets.....	2584
9.4.10 Zero touch of Tx (From Host) buffer descriptors.....	2585
9.4.11 Continuous transfer of packets from the queue with an end of transmission indication.....	2587
9.4.12 Controlled transmission of packets.....	2588
9.4.13 Egress Packet Operations.....	2588
9.4.14 IEEE 1588 Clock Synchronization Support.....	2590
9.4.15 Rate Limiting (Traffic Shaping) (P802.1Qav/D6.0).....	2595
9.4.16 Transmit Priority Escalation.....	2596
9.4.17 Enhanced Scheduled Traffic (EST – P802.1Qbv/D2.2).....	2596
9.4.18 DSCP.....	2599
9.4.19 Packet Priority Handling.....	2599
9.4.20 Packet CRC Handling.....	2601
9.4.21 Ethernet FIFO.....	2602
9.4.22 Ethernet Receive Flow Control.....	2602

9.4.23 Energy Efficient Ethernet Support (802.3az).....	2604
9.4.24 Switch Latency.....	2605
9.4.25 Emulation Control.....	2605
9.4.26 Software IDLE.....	2605
9.4.27 Common Platform Time Sync (CPTS).....	2605
9.4.28 MII Management Interface (MDIO).....	2622
9.4.29 Reset Isolation.....	2624
9.4.30 CPSW Initialization and Configuration.....	2625
9.4.31 Enet Mac Reset or XGMII/GMII Mode Change Configuration.....	2625
9.4.32 Memory Error Detection and Correction.....	2625
9.4.33 CPSW Network Statistics.....	2627
9.4.34 CPPI Streaming Packet Interface.....	2643
9.4.35 Egress Packet Operations.....	2649
9.5 MCU_CPSW0 Programming Guide.....	2649
9.5.1 Initialization and Configuration of CPSW Subsystem.....	2649
9.5.2 CPSW Reset.....	2650
9.5.3 MDIO Software Interface.....	2650
9.6 MSS_CPSW Registers.....	2652
<b>10 Quad Serial Peripheral Interface (QSPI).....</b>	<b>3365</b>
10.1 Quad Serial Peripheral Interface Overview.....	3366
10.2 QSPI Environment.....	3367
10.3 QSPI Functional Description.....	3368
10.3.1 QSPI Block Diagram.....	3368
10.3.2 QSPI Clock Configuration.....	3374
10.3.3 QSPI Interrupt Requests.....	3374
10.3.4 QSPI Memory Regions.....	3375
10.4 QSPI Registers.....	3377
10.4.1 QSPI Register Summary.....	3377
10.4.2 QSPI Register Description.....	3377
<b>11 Enhanced Direct Memory Access (EDMA).....</b>	<b>3389</b>
11.1 EDMA Module Overview.....	3389
11.1.1 EDMA Features.....	3390
11.2 EDMA Integration.....	3391
11.2.1 EDMA Integration.....	3392
11.2.2 EDMA Integration in MSS.....	3392
11.2.3 EDMA Integration in DSS.....	3394
11.2.4 EDMA Functional Interrupt Aggregator.....	3394
11.2.5 EDMA Error Interrupt Aggregator.....	3395
11.2.6 EDMA Configuration.....	3396
11.3 EDMA Controller Functional Description.....	3398
11.3.1 Block Diagram.....	3398
11.3.2 Types of EDMA Controller Transfers.....	3401
11.3.3 Parameter RAM (PaRAM).....	3403
11.3.4 Initiating a DMA Transfer.....	3417
11.3.5 Completion of a DMA Transfer.....	3420
11.3.6 Event, Channel, and PaRAM Mapping.....	3421
11.3.7 EDMA Channel Controller Regions.....	3423
11.3.8 Chaining EDMA Channels.....	3426
11.3.9 EDMA Interrupts.....	3427
11.3.10 Memory Protection.....	3436
11.3.11 Event Queue(s).....	3439
11.3.12 EDMA Transfer Controller (EDMA_TPTC).....	3441
11.3.13 Event Dataflow.....	3443
11.3.14 EDMA Controller Prioritization.....	3444
11.3.15 Emulation Considerations.....	3446
11.4 EDMA Transfer Examples.....	3447
11.4.1 Block Move Example.....	3447
11.4.2 Subframe Extraction Example.....	3449
11.4.3 Data Sorting Example.....	3450
11.4.4 Setting Up an EDMA Transfer.....	3452
11.5 EDMA Debug Checklist and Programming Tips.....	3454

11.5.1 EDMA Debug Checklist.....	3454
11.5.2 EDMA Programming Tips.....	3456
11.6 EDMA Event Map.....	3456
11.6.1 MSS TPCC_A Event Map.....	3456
11.6.2 MSS TPCC_B Event Map.....	3458
11.6.3 DSS TPCC_A Event Map.....	3460
11.6.4 DSS TPCC_B / TPCC_C Event Map.....	3461
11.6.5 Radar Subsystem (RSS TPCC_A) EDMA Event Map.....	3464
11.7 EDMA Request Map.....	3466
11.8 EDMA Register Manual.....	3467
11.8.1 EDMA Registers.....	3467
<b>12 Multi-Buffered Serial Peripheral Interface Module (MibSPI) with Parallel Pin Option (MibSPIP).....</b>	<b>3708</b>
12.1 Overview.....	3709
12.1.1 Word Format Options.....	3709
12.1.2 Multi-buffering (Mib) Support.....	3709
12.1.3 Transmission Lock (Multi-Buffer Mode Controller Only).....	3710
12.2 Operating Modes.....	3710
12.2.1 Pin Configurations.....	3711
12.2.2 Data Handling.....	3711
12.2.3 Operation with $\overline{\text{SPICS}}$ .....	3714
12.2.4 Operation with $\overline{\text{SPIENA}}$ .....	3715
12.2.5 Five-Pin Operation (Hardware Handshaking).....	3716
12.2.6 Data Formats.....	3717
12.2.7 Clocking Modes.....	3718
12.2.8 Data Transfer Example.....	3720
12.2.9 Decoded and Encoded Chip Select (Controller Only).....	3721
12.2.10 Variable Chip Select Setup and Hold Timing (Controller Only).....	3721
12.2.11 Hold Chip-Select Active.....	3721
12.2.12 Detection of Peripheral Desynchronization (Controller Only).....	3722
12.2.13 ENA Signal Time-Out (Controller Only).....	3723
12.2.14 Data-Length Error.....	3723
12.2.15 Parallel Mode (Multiple PICO/POCI Support, not available on all devices).....	3723
12.2.16 Continuous Self-Test (Controller/Peripheral).....	3731
12.2.17 Half Duplex Mode.....	3731
12.3 Test Features.....	3731
12.3.1 Internal Loop-Back Test Mode (Controller Only).....	3731
12.3.2 Input/Output Loopback Test Mode.....	3732
12.4 General-Purpose I/O.....	3733
12.5 Interrupts.....	3734
12.5.1 Interrupts in Multi-Buffer Mode.....	3734
12.6 DMA Interface.....	3737
12.6.1 DMA in Multi-Buffer Mode.....	3737
12.7 Module Configuration.....	3738
12.7.1 Compatibility (SPI) Mode Configuration.....	3738
12.7.2 MibSPI Mode Configuration.....	3739
12.8 Control Registers.....	3740
12.8.1 SPI Global Control Register 0 (SPIGCR0).....	3741
12.8.2 SPI Global Control Register 1 (SPIGCR1).....	3742
12.8.3 SPI Interrupt Register (SPIINT0).....	3743
12.8.4 SPI Interrupt Level Register (SPILVL).....	3746
12.8.5 SPI Flag Register (SPIFLG).....	3748
12.8.6 SPI Pin Control Register 0 (SPIPC0).....	3751
12.8.7 SPI Pin Control Register 1 (SPIPC1).....	3752
12.8.8 SPI Pin Control Register 2 (SPIPC2).....	3754
12.8.9 SPI Pin Control Register 3 (SPIPC3).....	3756
12.8.10 SPI Pin Control Register 4 (SPIPC4).....	3757
12.8.11 SPI Pin Control Register 5 (SPIPC5).....	3760
12.8.12 SPI Pin Control Register 6 (SPIPC6).....	3761
12.8.13 SPI Pin Control Register 7 (SPIPC7).....	3764
12.8.14 SPI Pin Control Register 8 (SPIPC8).....	3765
12.8.15 SPI Transmit Data Register 0 (SPIDAT0).....	3766

12.8.16 SPI Transmit Data Register 1 (SPIDAT1).....	3768
12.8.17 SPI Receive Buffer Register (SPIBUF).....	3771
12.8.18 SPI Emulation Register (SPIEMU).....	3774
12.8.19 SPI Delay Register (SPIDELAY).....	3774
12.8.20 SPI Default Chip Select Register (SPIDEF).....	3777
12.8.21 SPI Data Format Registers (SPIFMT).....	3778
12.8.22 Interrupt Vector 0 (INTVECT0).....	3781
12.8.24 Interrupt Vector 1 (INTVECT1).....	3782
12.8.25 SPI Pin Control Register 9 (SPIPC9).....	3784
12.8.26 Parallel/Modulo Mode Control Register (SPIPMCTRL).....	3785
12.8.27 Multi-buffer Mode Enable Register (MIBSPIE).....	3788
12.8.28 TG Interrupt Enable Set Register (TGITENST).....	3789
12.8.29 TG Interrupt Enable Clear Register (TGITENCR).....	3790
12.8.30 Transfer Group Interrupt Level Set Register (TGITLVST).....	3791
12.8.31 Transfer Group Interrupt Level Clear Register (TGITLVCR).....	3792
12.8.32 Transfer Group Interrupt Flag Register (TGINTFLG).....	3793
12.8.33 Tick Count Register (TICKCNT).....	3794
12.8.34 Last TG End Pointer (LTGPEND).....	3796
12.8.35 TGx Control Registers (TGxCTRL).....	3797
12.8.36 DMA Channel Control Register (DMAxCTRL).....	3802
12.8.37 DMAxCOUNT Register (ICOUNT).....	3805
12.8.38 DMA Large Count (DMACNTLEN).....	3806
12.8.39 Multi-buffer RAM Uncorrectable Parity Error Control Register (UERRCTRL).....	3806
12.8.40 Multi-buffer RAM Uncorrectable Parity Error Status Register (UERRSTAT).....	3808
12.8.41 RXRAM Uncorrectable Parity Error Address Register (UERRADDR1).....	3809
12.8.42 TXRAM Uncorrectable Parity Error Address Register (UERRADDR0).....	3810
12.8.43 RXRAM Overrun Buffer Address Register (RXOVRN_BUF_ADDR).....	3811
12.8.44 I/O-Loopback Test Control Register (IOLPBKTSTCR).....	3812
12.8.45 SPI Extended Prescale Register 1 (EXTENDED_PRESCALE1 for SPIFMT0 and SPIFMT1).....	3814
12.8.46 SPI Extended Prescale Register 2 (EXTENDED_PRESCALE2 for SPIFMT2 and SPIFMT3).....	3816
12.9 Multi-Buffer RAM.....	3818
12.9.1 Multi-Buffer RAM Auto Initialization.....	3819
12.9.2 Multi-Buffer RAM Register Summary.....	3819
12.9.3 Multi-Buffer RAM Transmit Data Register (TXRAM).....	3820
12.9.4 Multi-Buffer RAM Receive Buffer Register (RXRAM).....	3823
12.10 Parity Memory.....	3826
12.10.1 Example of Parity Memory Organization.....	3828
12.11 MibSPI Pin Timing Parameters.....	3829
12.11.1 Controller Mode Timings for SPI/MibSPI.....	3829
12.11.2 Peripheral Mode Timings for SPI/MibSPI.....	3831
12.11.3 Controller Timing Parameter Details.....	3832
12.11.4 Peripheral Mode Timing Parameter Details.....	3832
<b>13 Vectored Interrupt Manager (VIM) Module.....</b>	<b>3833</b>
13.1 VIM Overview.....	3834
13.2 VIM Interrupt Inputs.....	3834
13.3 VIM Interrupt Outputs.....	3834
13.4 VIM Interrupt Vector Table (VIM RAM).....	3834
13.5 VIM Interrupt Prioritization.....	3835
13.6 VIM ECC Support.....	3835
13.7 VIM Lockstep Mode.....	3836
13.8 VIM IDLE State.....	3836
13.9 VIM Interrupt Handling.....	3836
13.9.1 Servicing IRQ Through Vector Interface.....	3836
13.9.2 Servicing IRQ Through MMR Interface.....	3837
13.9.3 Servicing IRQ Through MMR Interface (Alternative).....	3837
13.9.4 Servicing FIQ.....	3838
13.9.5 Servicing FIQ (Alternative).....	3838
13.10 MSS_VIM Registers.....	3840
<b>14 Real-Time Interrupt (RTI) and Watchdog Module.....</b>	<b>4441</b>
14.1 Overview.....	4441
14.1.1 Features.....	4441

14.1.2 Industry Standard Compliance Statement.....	4441
14.2 Module Operation.....	4442
14.2.1 Counter Operation.....	4442
14.2.2 Interrupt/DMA Requests.....	4444
14.2.3 RTI Clocking.....	4445
14.2.4 Digital Watchdog (DWD).....	4445
14.2.5 Halting Debug Mode Behaviour.....	4449
14.3 MSS_RTIC Registers.....	4450
<b>15 UART/SCI.....</b>	<b>4494</b>
15.1 Introduction.....	4494
15.1.1 SCI Features.....	4494
15.1.2 Block Diagram.....	4494
15.2 SCI Communication Formats.....	4497
15.2.1 SCI Frame Formats.....	4497
15.2.2 SCI Timing Mode.....	4497
15.2.3 SCI Baud Rate.....	4499
15.2.4 SCI Multiprocessor Communication Modes.....	4499
15.3 SCI Interrupts.....	4502
15.3.1 Transmit Interrupt.....	4503
15.3.2 Receive Interrupt.....	4503
15.3.3 WakeUp Interrupt.....	4503
15.3.4 Error Interrupts.....	4504
15.4 SCI DMA Interface.....	4505
15.4.1 Receive DMA Requests.....	4505
15.4.2 Transmit DMA Requests.....	4505
15.5 SCI Configurations.....	4507
15.5.1 Receiving Data.....	4507
15.5.2 Transmitting Data.....	4508
15.6 MSS_SCI Registers.....	4509
15.7 SCI GPIO Functionality.....	4550
15.7.1 GPIO Functionality.....	4550
15.7.2 Under Reset.....	4550
15.7.3 Out of Reset.....	4551
15.7.4 Open-Drain Feature Enabled on a Pin.....	4551
15.7.5 Summary.....	4551
<b>16 Inter-Integrated Circuit (I2C) Module.....</b>	<b>4552</b>
16.1 Overview.....	4552
16.1.1 Introduction to the I2C Module.....	4553
16.1.2 Functional Overview.....	4553
16.1.3 Clock Generation.....	4556
16.2 I2C Module Operation.....	4557
16.2.1 Input and Output Voltage Levels.....	4557
16.2.2 I2C Module Reset Conditions.....	4557
16.2.3 I2C Module Data Validity.....	4557
16.2.4 I2C Module Start and Stop Conditions.....	4558
16.2.5 Serial Data Formats.....	4558
16.2.6 NACK Bit Generation.....	4560
16.3 I2C Operation Modes.....	4561
16.3.1 Controller Transmitter Mode.....	4561
16.3.2 Controller Receiver Mode.....	4561
16.3.3 Target Transmitter Mode.....	4561
16.3.4 Target Receiver Mode.....	4561
16.3.5 Free Run Mode.....	4562
16.3.6 Ignore NACK Mode.....	4562
16.4 I2C Module Integrity.....	4563
16.4.1 Arbitration.....	4563
16.4.2 I2C Clock Generation and Synchronization.....	4564
16.4.3 Prescaler.....	4564
16.4.4 Noise Filter.....	4564
16.5 Operational Information.....	4565
16.5.1 I2C Module Interrupts.....	4565



16.5.2 DMA Controller Events.....	4565
16.5.3 I2C Enable/Disable.....	4566
16.5.4 General Purpose I/O.....	4566
16.5.5 Pull Up/Pull Down Function.....	4567
16.5.6 Open Drain Function.....	4567
16.6 MSS_I2C Registers.....	4568
16.7 Sample Waveforms.....	4597
<b>17 General-Purpose Input/Output (GIO) Module.....</b>	<b>4598</b>
17.1 Overview.....	4598
17.2 Quick Start Guide.....	4599
17.3 Functional Description of GIO Module.....	4601
17.3.1 I/O Functions.....	4601
17.3.2 Interrupt Function.....	4602
17.3.3 GIO Block Diagram.....	4603
17.4 Device Modes of Operation.....	4604
17.4.1 Emulation Mode.....	4604
17.4.2 Power-Down Mode (Low-Power Mode).....	4604
17.4.3 Interrupts.....	4604
17.5 TOP_GIO Registers.....	4605
17.6 I/O Control Summary.....	4693
<b>18 Enhanced Pulse Width Modulator (ePWM) Module.....</b>	<b>4694</b>
18.1 Introduction.....	4695
18.1.1 Submodule Overview.....	4695
18.1.2 Register Mapping.....	4698
18.2 ePWM Submodules.....	4699
18.2.1 Overview.....	4699
18.2.2 Time-Base (TB) Submodule.....	4701
18.2.3 Counter-Compare (CC) Submodule.....	4712
18.2.4 Action-Qualifier (AQ) Submodule.....	4718
18.2.5 Dead-Band Generator (DB) Submodule.....	4731
18.2.6 PWM-Chopper (PC) Submodule.....	4737
18.2.7 Trip-Zone (TZ) Submodule.....	4741
18.2.8 Event-Trigger (ET) Submodule.....	4748
18.2.9 Digital Compare (DC) Submodule.....	4753
18.2.10 Proper Interrupt Initialization Procedure.....	4759
18.3 Application Examples.....	4759
18.3.1 Overview of Multiple Modules.....	4759
18.3.2 Key Configuration Capabilities.....	4760
18.3.3 Controlling Multiple Buck Converters With Independent Frequencies.....	4761
18.3.4 Controlling Multiple Buck Converters With Same Frequencies.....	4764
18.3.5 Controlling Multiple Half H-Bridge (HHB) Converters.....	4767
18.3.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM).....	4769
18.3.7 Practical Applications Using Phase Control Between PWM Modules.....	4772
18.4 ePWM Module Control and Status Registers.....	4774
<b>19 On-chip Debug.....</b>	<b>4824</b>
19.1 AWR2544 DebugSS Architecture .....	4825
19.1.1 Overview.....	4825
19.1.2 DebugSS Architecture .....	4825
19.1.3 Debug Subsystem Address Map .....	4826
19.1.4 Trace Subsystem.....	4827
19.1.5 Suspend Peripherals.....	4830
<b>20 Safety Modules.....</b>	<b>4831</b>
20.1 Core Clock Comparator (CCC).....	4832
20.1.1 Description.....	4832
20.1.2 Block Diagram.....	4832
20.1.3 Perform Clock Comparison.....	4833
20.1.4 I/O Table.....	4833
20.1.5 Recommended Programming.....	4833
20.2 Dual Clock Comparator (DCC).....	4833
20.2.1 Introduction.....	4834
20.2.2 Module Operation.....	4835

20.2.3 MSS DCC Integration.....	4839
20.2.4 Clock Source Selection for Counter0 and Counter1.....	4840
20.2.5 DCC Registers.....	4841
20.3 ECC_AGGREGATOR.....	4852
20.3.1 ECC Aggregator Overview.....	4852
20.3.2 Integration Details.....	4852
20.3.3 ECC Aggregator Features.....	4852
20.3.4 ECC Aggregator Integration.....	4853
20.3.5 ECC Aggregator Function Description.....	4854
20.3.6 ECC Aggregator Register Groups.....	4855
20.3.7 Read Access to the ECC Control and Status Registers.....	4855
20.3.8 Serial Write Operation.....	4855
20.3.9 Interrupts.....	4856
20.3.10 Inject Only Mode.....	4856
20.3.11 Errors.....	4857
20.3.12 Aggregator Mapping to Memory Instances.....	4857
20.4 ECC Registers.....	4859
20.4.1 DSS_ECC_AGG Registers.....	4860
20.4.2 MSS_ECC_AGGA Registers.....	4884
20.4.3 MSS_ECC_AGGB Registers.....	4913
20.4.4 MSS_ECC_AGG_MSS Registers.....	4942
20.5 Error Signaling Module (ESM).....	4964
20.5.1 Overview.....	4964
20.5.2 Feature List.....	4964
20.5.3 Block Diagram.....	4965
20.5.4 Integration Details.....	4966
20.5.5 Module Operation.....	4968
20.5.6 Recommended Programming Procedure.....	4973
20.5.7 Main Subsystem ESM Interrupt Map.....	4973
20.5.8 MSS_ESM Registers.....	4979
20.6 Cyclic Redudancy Check (CRC).....	5010
20.6.1 Overview.....	5010
20.6.2 Features.....	5010
20.6.3 Block Diagram.....	5010
20.6.4 Module Operation.....	5012
20.6.5 Example.....	5022
20.6.6 MSS_MCRC Registers.....	5025
20.7 Self-Test Controller (STC).....	5094
20.7.1 Integration Spec.....	5094
20.7.2 STC Overview.....	5094
20.7.3 Block Diagram.....	5095
20.7.4 Module Description.....	5096
20.7.5 STC Registers.....	5106
20.8 Programmable Built-In Self-Test (PBIST).....	5178
20.8.1 Overview.....	5178
20.8.2 RAM/ROM Grouping and Algorithm.....	5180
<b>21 GPADC.....</b>	<b>5182</b>
21.1 GPADC Overview.....	5182
21.2 Functional Description Modes of Operation.....	5183
21.2.1 IFM Mode (Inter Frame Monitoring Mode).....	5183
21.3 Interrupts.....	5183
21.4 GPADC Trigger Sources.....	5183
21.4.1 IFM Mode.....	5183
21.5 MSS GPADC Register Specification.....	5183
21.5.1 MSS_GPADC_REG Registers.....	5184
<b>22 LVDS.....</b>	<b>5208</b>
22.1 LVDS Overview.....	5209
22.2 LVDS Programming Sequence.....	5209
22.2.1 LVDS Global Initialization.....	5209
22.2.2 CBUFF Configuration.....	5209
22.3 CBUFF and LVDS Registers.....	5211



22.3.1 DSS_CBUFF Registers.....	5212
<b>23 Mailbox.....</b>	<b>5359</b>
23.1 Mailbox Message Scheme.....	5361
<b>24 Radar Hardware Accelerator 1.5 User Guide.....</b>	<b>5363</b>
24.1 Radar Hardware Accelerator 1.5 User Guide – Part 1.....	5364
24.1.1 Radar Hardware Accelerator – Overview.....	5364
24.1.2 Accelerator Engine – State Machine.....	5371
24.1.3 Accelerator Engine – Input Formatter.....	5377
24.1.4 Accelerator Engine – Output Formatter.....	5383
24.1.5 Accelerator Engine – Core Computational Unit.....	5389
24.1.6 Appendix A.....	5397
24.2 Radar Hardware Accelerator 1.5 User Guide – Part 2.....	5398
24.2.1 FFT Engine – Pre-Processing.....	5398
24.3 Radar Hardware Accelerator 1.5 User Guide – Part 3: Compression Engine.....	5419
24.3.1 Compression/Decompression Engine Overview.....	5420
24.3.2 Algorithms.....	5420
24.3.3 Operation.....	5422
<b>25 Glossary.....</b>	<b>5429</b>
<b>26 Revision History.....</b>	<b>5433</b>

This page intentionally left blank.



The AWR2544 device is targetted to serve an evolving Satellite vehicle architecture wherein the near raw radar sensor data (compressed 1D FFT output) is sent to a central compute engine over a high speed interface. The AWR2544 is a single chip radar transceiver with integrated 76-81 GHz LO, 4 RXs, and 4 TXs, with 1Gbps Ethernet capability for streaming data at high speed, along with an integrated main MCU that is accessible for customer use. This chapter introduces the features, subsystems, and architecture of AWR2544 Systems on Chip (SoCs).

### 1.1 Device Overview

The AWR2544 is targeted for Automotive Satellite Radar Sensor System. This SoC aims to meet the complex processing needs of end equipment such as High End corner Radar, Entry level/Mid end Front Radar, and Blind Spot Detection (BSD).

The SoC has been designed as a high performance and highly integrated device architecture, adding significant enhancement on processing power and coherent memory support. In addition, they support state of art security and functional safety features.

Some of the main distinguished characteristics of the device are:

- Frequency-modulated Continuous Wave (FMCW) Radio Frequency Transceiver
  - Integrated PLL, Transmitter, Receiver, Baseband and A2D
  - 76-81 GHz Coverage with 5GHz available Bandwidth
  - 4 Receive channels
  - 4 Transmit channels
  - Per Transmit Phase Shifter
  - Ultra-accurate Chirp engine based on Fractional-n PLL
- Wide IF bandwidth up to 20MHz to decrease range/Doppler ambiguity
- Highly programmable chirp generation
- 12, 14, and 16-bit I-Only ADC With Variable Baseband ADC Sampling Rates up to 37.5 Msps.
- High Performance Radar Processing Accelerators having capability to offload the CPU almost all the pre-processing functions.
- Hardware Security Module (HSM) – To provide secure execution environment.
- High Performance/Cost ratio. Highly optimized, area efficient infrastructure to support the radar data flow with maximum cost effectiveness and efficiency.
- High Temperature operating range ( $>125^{\circ}\text{C}$ ) for automotive use cases.
- 2 Mbytes of 'On Chip' RAM.

The device is composed of the following main subsystems:

- One Lock-step Cortex R5F microcontroller at up to 300 MHz.
- Radar Hardware Accelerators running at 300 MHz
- Cortex M4-based () Hardware Security Module (HSM) for running security services in a secure island.
- Radar Subsystem (RSS) comprises of a lockstep CR4 microcontroller at up to 200 MHz, Digital front end (DFE) at up to 200 MHz, and Analog-RF subsystem. This group of modules is referred as BIST SubSystem(BSS) and in the later chapters BSS or RSS will be used interchangeably as and when required.

### Note

The R5F core of AWR2544 devices only supports lock-step mode.

The device provides a rich set of peripherals, such as:

- General connectivity peripherals, including:
  - One Inter-Integrated Circuit (I2C) interface
  - Two Controller/Peripheral Multi-Buffered Serial Peripheral Interfaces (MiBSPI)
  - Two configurable Universal Asynchronous Receiver/Transmitter (UART) interfaces
  - One General-Purpose Input/Output (GPIO) module
- High-speed interfaces, including:
  - One 1 Gbps Ethernet switch (CPSW)
- Output Control interface
  - Three Enhanced Pulse Width Modulation (EPWM) modules
- EMIF Interface
  - One Quad-Serial Peripheral Interface (QSPI) at up to 75 MHz.
- Timers and Watchdog Module
  - Three Real Time Interrupts (RTI) modules in MSS
  - One Watchdog modules in MSS (Same RTI IP but in Watchdog configuration)
- Interprocessor Communication (IPC) interface
  - Mailbox module for interprocessor communication between the different modules
- Sub-system Reset and control module with device top-level configurations:
  - Power distribution, reset controls, and clock management components.
  - Registers for the following functions:
    - I/O Configurations
    - PLL control and associated High-speed Dividers (HSDIV)
    - Clock Selection
    - System boot mode decoding logic
  - Up to 2MB of on-chip memory split across MSS, and the shared memory L3 bank.
  - Debug and trace capabilities.

The device includes different modules for functional safety requirements support:

- Logic BIST mechanism for all the CPU cores
- PBIST mechanism for all the memories
- ECC on the critical memories
- MPU on all critical shared resources – MMRs and memories
- Voltage monitor on all the primary supplies with >90% DC
- Clock monitors to monitor all the primary clocks. At least one Watchdog per each subsystem.
- PLL Lock monitors – PHASELOCK, FREQLOCK
- Temperature sensors with an accuracy of +/-5°C near R5F, HSM, and all other temperature sensitive locations.
- Separate safety island (MCU) with Lock-Step Cortex-R5F.
- Safety enabled interconnect
- One Error Signalling Module (ESM) to enable error monitoring
- Temperature monitoring sensors
- Dedicated hardware Memory Cyclic Redundancy Check (MCRC) blocks.

## 1.2 Device Block Diagram

### AWR2544 – Block Diagram

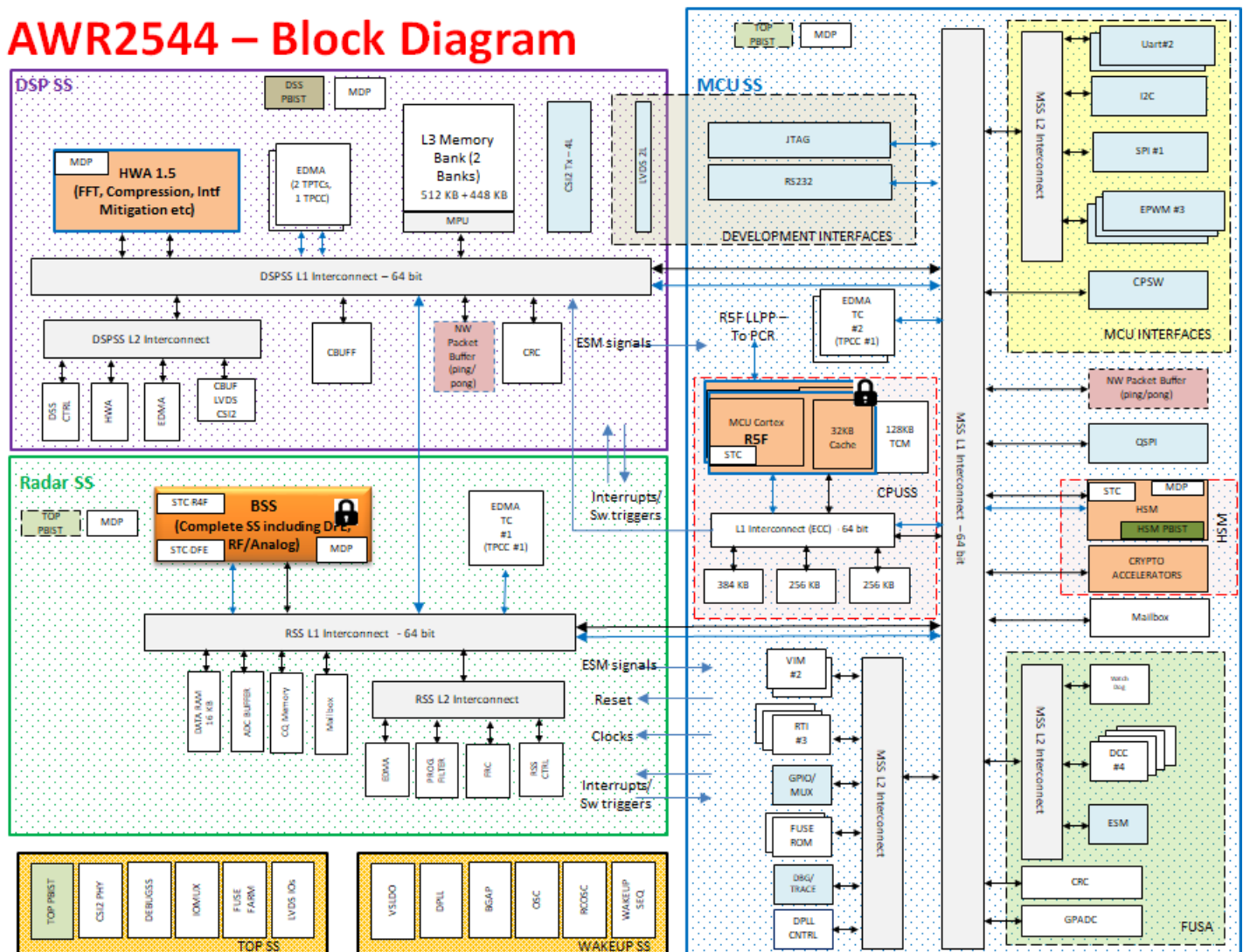


Figure 1-1. Block Diagram

Table 1-1. Module Allocation and Instances within the Device

Module Full Name	Module Abbreviation	MSS	RSS
ARM® Lockstep R5F Subsystem	R5FSS	1	
ARM Cortex® M4 Hardware Security Module Subsystem	HSM	1	
Common Platform Ethernet Switch	CPSW	1	
Quad-Serial Peripheral Interface	QSPI	1	
Enhanced Direct Memory Access	EDMA_TPCC EDMA_TPTC	TPCC- 1 TPTC-2	TPCC- 1 TPTC-1
Serial Peripheral Interface	MIBSPI	1	
Vectored Interrupt Manager	VIM	2	
Real Time Interrupt Timer	RTI	3	

**Table 1-1. Module Allocation and Instances within the Device (continued)**

Module Full Name	Module Abbreviation	MSS	RSS
Windowed Watchdog Timer	RTI-WWDT	1	
Universal Asynchronous Receiver/ Transmitter	UART	2	1
RS232	RS232	1	
Inter Integrated Circuit	I2C	1	
General Purpose Input Output	GPIO	1	
EFUSE Controller	EFUSE	2	
Debug Subsystem	DEBUGSS	1	
Dual Clock Comparator	DCC	4	
Cyclic Redundancy Check	CRC	1	
General Purpose Analog to Digital Convertor	GPADC	1	1
Enhanced Pulse Width Modulation	EPWM	3	
Error Signaling Module	ESM	1	
Radar Accelerators	HWA1.5		
LVDS Interface	LVDS		
Interprocess Communication	MAILBOX	1	1

Note: Above table provides module/components list for all the subsystem except BSS. Bist SubSystem (BSS), part of RadarSS (RSS) is a blackbox and not available for user.



**Table 2-1. AWR2544 Memory Map**

Device Variant	DSS L3	MSS L2	MSS L1
AWR2544	960KB	896KB [384KB+256KB+256KB memory bank]	L1I:16KB L1D: 16KB TCM 128KB (TCMA: 64KB, TCMB: 64KB)

L3 consists of different memory banks over which Bank3 is allocated to MSS\_L2 or kept with itself.

Memory Bank	Device (in KB)
Bank 0	512
Bank 1	448
Bank 2	DSS_L3 = 0 / MSS_L2 = 256 DSS_L3 = 256 / MSS_L2 = 0

**Note**

There is no restriction fundamentally on access to MSS and DSS memories. The address regions (global view) need to be appropriately selected. There are MPUs at the key memories (for ex., MSS L2, DSS L3,...) for application to restrict the access as needed.

## 2.1 Main Subsystem Cortex R5F Memory Map

Module Name	Base Address	Size
TCMA_ROM_CR5A	0x0000 0000	128 KBytes
TCMA_RAM_CR5A	0x0002 0000	64 KBytes
TCMB_CR5A	0x0008 0000	64 KBytes
MSS_SPIB_RAM	0x0202 0000	256Bytes
MSS_VIM_R5A	0x0208 0000	9 KBytes
MSS_IOMUX	0x020C 0000	512Bytes
MSS_RCM	0x0210 0000	4132 Bytes
MSS_CTRL	0x0212 0000	4132 Bytes
MSS_TOPRCM	0x0214 0000	4 KBytes
MSS_DEBUGSS	0x02A0 0000	248 KBytes
MSS_PCR1	0x02F7 8000	1 KBytes
TOP_PBIST	0x02F7 9400	464Bytes
MSS_R5SS_STC	0x02F7 9800	284Bytes
MSS_DCCA	0x02F7 9C00	60Bytes
MSS_DCCB	0x02F7 9D00	60Bytes

MSS_DCCC	0x02F7 9E00	60Bytes
MSS_DCCD	0x02F7 9F00	60Bytes
MSS_RTIA	0x02F7 A000	192Bytes
MSS_RTIB	0x02F7 A100	192Bytes
MSS_RTIC	0x02F7 A200	192Bytes
MSS_WDT	0x02F7 A300	192Bytes
MSS_ESM	0x02F7 A400	220Bytes
TOP_EFUSE_FARM	0x02F7 A800	64Bytes
MSS_CCMR	0x02F7 AC00	28Bytes
MSS_I2C	0x02F7 B000	100Bytes
MSS_GIO	0x02F7 B400	341Bytes
MSS_ECC_AGG_R5A	0x02F7 B800	528Bytes
MSS_ECC_AGG_MSS	0x02F7 C000	528Bytes
MSS_SPIB	0x02F7 EA00	512Bytes
MSS_SCIA	0x02F7 EC00	148Bytes
MSS_SCIB	0x02F7 ED00	148Bytes
MSS_GPADC_PKT_RAM	0x030C 0000	2 KBytes
TOP_CTRL	0x030E 0000	4 KBytes
MSS_TPCC_A	0x0310 0000	16 KBytes
MSS_TPTC_A0	0x0314 0000	860Bytes
MSS_TPTC_A1	0x0316 0000	860Bytes
MSS_PCR2	0x03F7 8000	1 KBytes
MSS_ETPWMA	0x03F7 8C00	116Bytes
MSS_ETPWMB	0x03F7 8D00	116Bytes
MSS_ETPWMC	0x03F7 8E00	116Bytes
MSS_GPADC_REG	0x03F7 9800	92Bytes
RSS_RCM	0x0500 0000	4 KBytes
RSS_CTRL	0x0502 0000	4 KBytes
RSS_PROC_CTRL	0x0520 0000	4 KBytes
RSS_PROC_ECC_AGG	0x0522 0000	528Bytes
RSS_PCR	0x05F7 8000	1KBytes
DSS_RCM	0x0600 0000	4132 Bytes
DSS_CTRL	0x0602 0000	4132 Bytes
DSS_CBUFF	0x0604 0000	564 Bytes
DSS_HWA_PARAM	0x0606 0000	4KBytes
DSS_HWA_CFG	0x0606 2000	4KBytes
DSS_HWA_WINDOW_RAM	0x0606 4000	8KBytes
DSS_ECC_AGG	0x060A 0000	528 Bytes
DSS_TPTC_A	0x0610 0000	16KBytes
DSS_TPTC_A0	0x0616 0000	860 Bytes
DSS_TPTC_A1	0x0618 0000	860 Bytes
DSS_PCR	0x06F7 8000	1KBytes
DSS_DSP_PBIIST	0x06F7 9000	464 Bytes
MSS_CPSW	0x0700 0000	252 KBytes
MSS_L2	0x1020 0000	896KBytes
HSM_RAM	0x2002 0000	192KBytes
MPU_MSS_L2_BANKA	0x4002 0000	780Bytes



MPU_MSS_L2_BANKB	0x4004 0000	780Bytes
MPU_HSM_DTHER	0x4006 0000	780Bytes
MPU_MSS_MBOX	0x4008 0000	780Bytes
MPU_MSS_PCRA	0x400A 0000	780Bytes
MPU_MSS_QSPI	0x400C 0000	780Bytes
MPU_MSS_CR5A_AXIS	0x400E 0000	780Bytes
MPU_DSS_L3_BANKA	0x4012 0000	780Bytes
MPU_DSS_L3_BANKB	0x4014 0000	780Bytes
MPU_DSS_L3_BANKC	0x4016 0000	780Bytes
MPU_DSS_HWA_DMA0	0x401A 0000	780Bytes
MPU_DSS_HWA_DMA1	0x401C 0000	780Bytes
DSS_L3	0x8800 0000	960 KBytes
BSS_TCMB	0xA100 0000	64KBytes
RSS_ADCBUF_WRITE	0xA400 0000	16 KBytes
RSS_CHIRP_INFO_WRITE	0xA401 0000	8 KBytes
RSS_STATIC_MEM	0xA402 0000	16 KBytes
RSS_CR4_MBOX	0xA403 0000	8 KBytes
RSS_ADCBUF_READ	0xA500 0000	16 KBytes
RSS_CHIRP_INFO_READ	0xA501 0000	8 KBytes
MSS_TCMA_CR5A	0xC100 0000	192 KBytes*
MSS_TCMB_CR5A	0xC180 0000	64 KBytes
MSS_MBOX	0xC500 0000	8 KBytes
MSS_RETRAM	0xC501 0000	2 KBytes
MSS_MCRC	0xC502 0000	328Bytes
MSS_GPADC_DATA_RAM	0xC503 0000	2 KBytes
EXT_FLASH	0xC600 0000	32 MBytes
MSS_QSPI	0xC800 0000	116Bytes

### Note

For MSS\_TCMA\_CR5A 0xC100 0000, 192 KB size includes size of TCMA RAM (64kb) and TCMA ROM (128KB). ROM-to-RAM (RBL to SBL) switching causes TCMA\_ROM\_CR5A eclipse to TCMA\_RAM\_CR5A i.e. 0x0000 0000 now maps to TCMA\_RAM\_CR5A memory area.

### Note

TCMB are 0x0002 0000 and 0x0008 0000 respectively as per R5A view. For AXI slave (EDMA or other processors) view addresses are 0xC100 0000 and 0xC180 0000.

## 2.2 Radar Subsystem Memory Map

Module Name	Base Address	Size
DSS_L3	0x88000000	960KBytes
MSS_SPIB_RAM	0x52020000	256Bytes
MSS_VIM_R5A	0x52080000	9 KBytes
MSS_IOMUX	0x520C0000	512Bytes
MSS_RCM	0x52100000	4 KBytes
MSS_CTRL	0x52120000	4 KBytes

MSS_TOPRCM	0x52140000	4 KBytes
MSS_DEBUGSS	0x52A00000	248 KBytes
MSS_PCR1	0x52F78000	1 KBytes
TOP_PBIST	0x52F79400	464Bytes
MSS_R5SS_STC	0x52F79800	284Bytes
MSS_DCCA	0x52F79C00	44Bytes
MSS_DCCB	0x52F79D00	44Bytes
MSS_DCCC	0x52F79E00	44Bytes
MSS_DCCD	0x52F79F00	44Bytes
MSS_RTIA	0x52F7A000	192Bytes
MSS_RTIB	0x52F7A100	192Bytes
MSS_RTIC	0x52F7A200	192Bytes
MSS_WDT	0x52F7A300	192Bytes
MSS_ESM	0x52F7A400	224Bytes
TOP_EFUSE_FARM	0x52F7A800	64Bytes
MSS_CCMR	0x52F7AC00	28Bytes
MSS_I2C	0x52F7B000	100Bytes
MSS_GPIO	0x52F7B400	341Bytes
MSS_ECC_AGG_R5A	0x52F7B800	528Bytes
MSS_ECC_AGG_MSS	0x52F7C000	528Bytes
MSS_SPIB	0x52F7EA00	512Bytes
MSS_SCIA	0x52F7EC00	148Bytes
MSS_SCIB	0x52F7ED00	148Bytes
MSS_GPADC_PKT_RAM	0x530C0000	2 KBytes
TOP_CTRL	0x530E0000	4 KBytes
MSS_TPCC_A	0x53100000	16 KBytes
MSS_TPTC_A0	0x53140000	860Bytes
MSS_TPTC_A1	0x53160000	860Bytes
MSS_PCR2	0x53F78000	1 KBytes
MSS_ETPWMA	0x53F78C00	116Bytes
MSS_ETPWMB	0x53F78D00	116Bytes
MSS_ETPWMC	0x53F78E00	116Bytes
MSS_GPADC_REG	0x53F79800	92Bytes
RSS_RCM	0x55000000	4 KBytes
RSS_CTRL	0x55020000	4 KBytes
RSS_PROG_FILT	0x550A0000	228Bytes
MPU_RSS_MSS2RSS	0x550E0000	780Bytes
RSS_TPCC_A	0x55100000	16 KBytes
RSS_TPTC_A0	0x55160000	860Bytes
RSS_ECC_AGG	0x551C0000	528Bytes
RSS_FIR_COEFF	0x551E0000	1KBytes
RSS_FIR_DMEN	0x551E0400	768Bytes
RSS_PROC_CTRL	0x55200000	4 KBytes
RSS_PROC_ECC_AGG	0x55220000	528Bytes
RSS_PCR	0x55F78000	1 KBytes
DSS_RCM	0x56000000	4132 Bytes
DSS_CTRL	0x56020000	4132 Bytes

DSS_CBUFF	0x56040000	564 Bytes
DSS_HWA_PARAM	0x56060000	4KBytes
DSS_HWA_CFG	0x56062000	4KBytes
DSS_HWA_WINDOW_RAM	0x56064000	8KBytes
DSS_ECC_AGG	0x560A0000	528 Bytes
DSS_TPTC_A	0x56100000	16KBytes
DSS_TPTC_A0	0x56160000	860 Bytes
DSS_TPTC_A1	0x56180000	860 Bytes
DSS_PCR	0x56F78000	1KBytes
DSS_DSP_PBIIST	0x56F79000	464Bytes
MPU_MSS_L2_BANKA	0x40020000	780Bytes
MPU_MSS_L2_BANKB	0x40040000	780Bytes
MPU_HSM_DTHE	0x40060000	780Bytes
MPU_MSS_MBOX	0x40080000	780Bytes
MPU_MSS_PCRA	0x400A0000	780Bytes
MPU_MSS_QSPI	0x400C0000	780Bytes
MPU_MSS_CR5A_AXIS	0x400E0000	780Bytes
MPU_DSS_L3_BANKA	0x40120000	780Bytes
MPU_DSS_L3_BANKB	0x40140000	780Bytes
MPU_DSS_L3_BANKC	0x40160000	780Bytes
MPU_DSS_HWA_DMA0	0x401A0000	780Bytes
MPU_DSS_HWA_DMA1	0x401C0000	780Bytes
RSS_ADCBUG_WRITE	0xA400 0000	16 KBytes
RSS_CHIRP_INFO_WRITE	0xA401 0000	8 KBytes
RSS_STATIC_MEM	0xA402 0000	16 KBytes
RSS_ADCBUF_READ	0xA500 0000	16 KBytes
RSS_CHIRP_INFO_READ (Chirp Quality CQ data)	0xA501 0000	8 KBytes
MSS_L2	0xC0200000	896 KBytes
MSS_TCMA_CR5A	0xC1000000	192 KBytes*
MSS_TCMB_CR5A	0xC1800000	64 KBytes
MSS_MBOX	0xC5000000	8 KBytes
MSS_MCRC	0xC5020000	328Bytes
MSS_GPADC_DATA_RAM	0xC5030000	2 KBytes
EXT_FLASH	0xC6000000	32 MBytes
MSS_QSPI	0xC8000000	116Bytes

### Note

For MSS\_TCMA\_CR5A 0xC100 0000, 192 KB size includes size of TCMA RAM (64kb) and TCMA ROM (128KB).



The device implements a system interconnect based on TI's common bus architecture, comprising of VBUSM and VBUSP protocols.

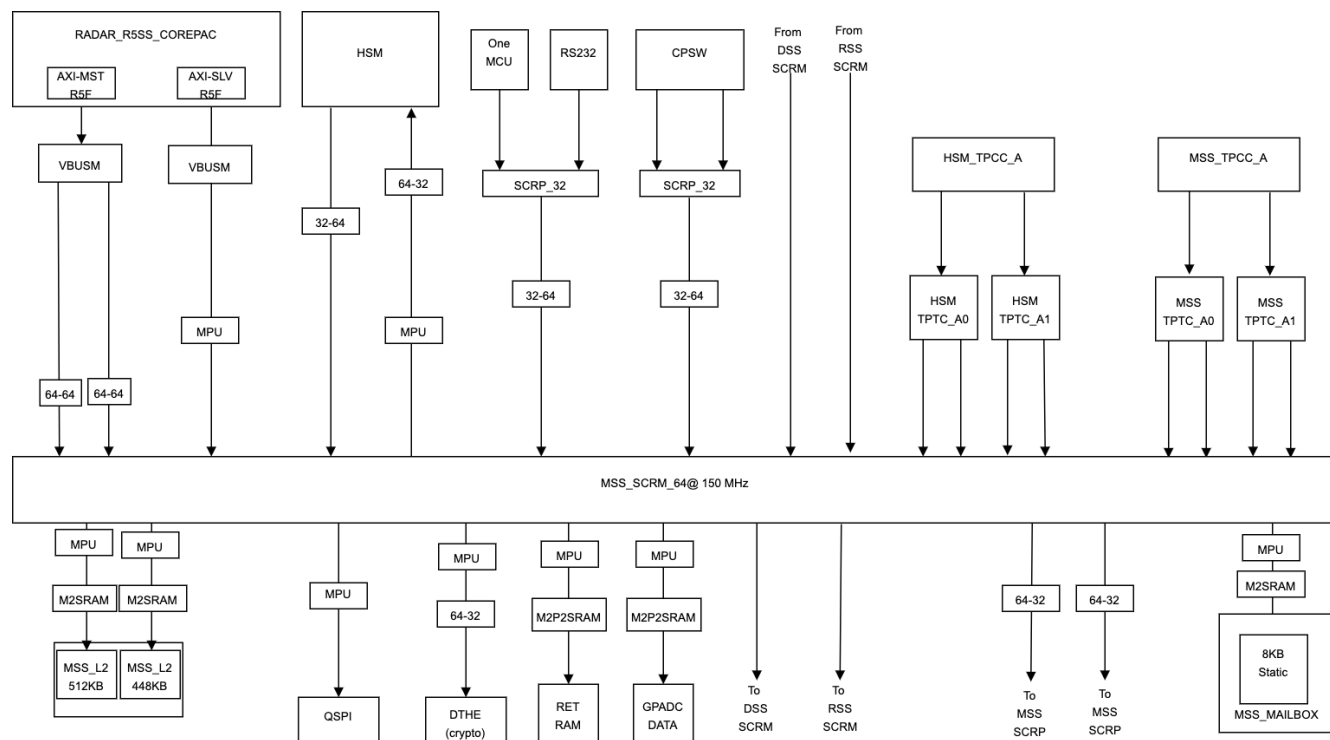
The system interconnect is designed for the high-performance needs of the system. The interconnect structure is a full crossbar implementation, wherein every controller has an independent communication path with every target such that transactions from each controllers have access to full interconnect bandwidth. Arbitration only happens at target end point. It's divided into interconnect systems local to each subsystem: the Main R5F subsystem, and Radar Subsystem .

### **3.1 Main Subsystem R5F Infrastructure**

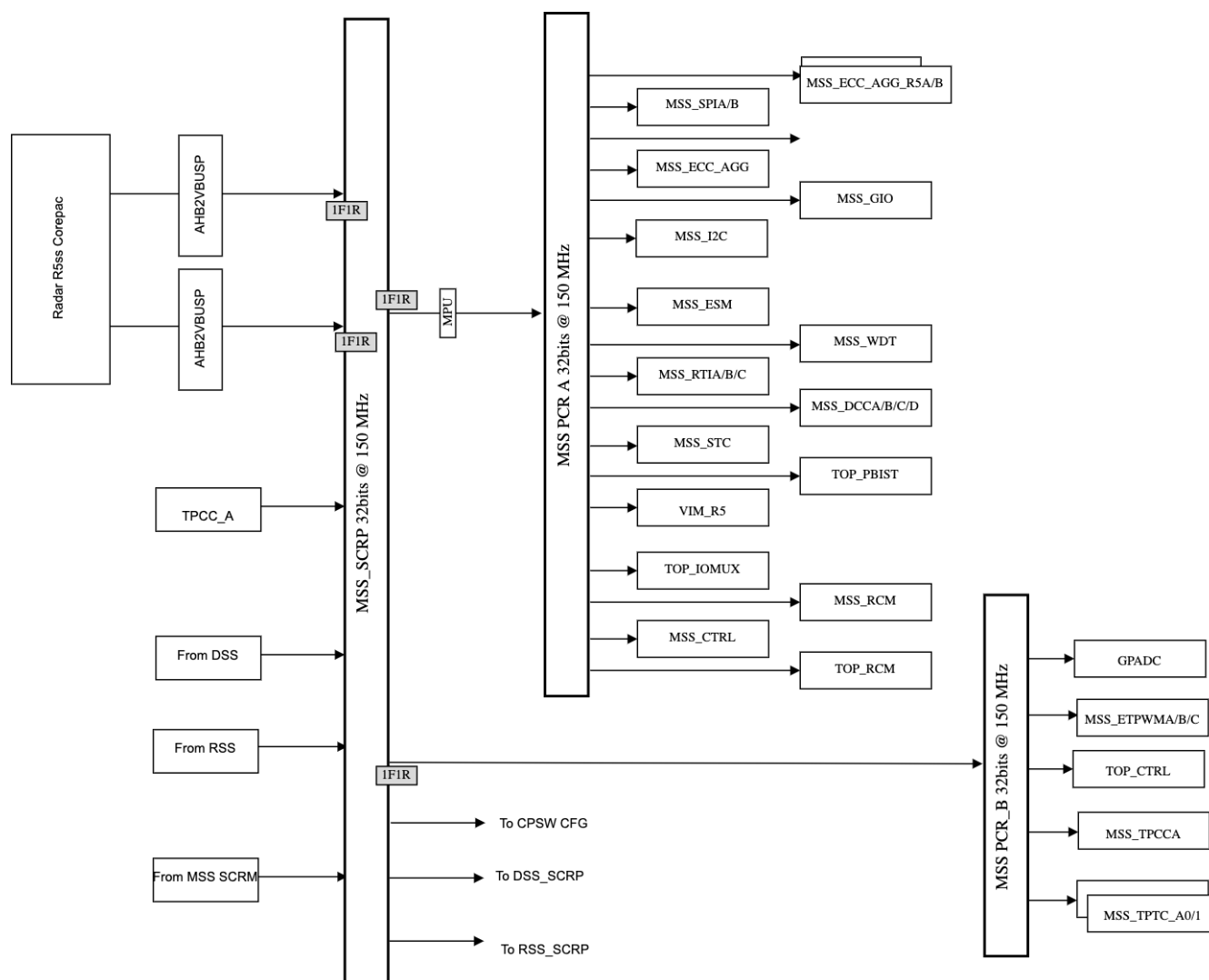
In the main subsystem, the primary VBUSM SCR is responsible for managing the arbitration priority between accesses from multiple controllers to each of the targets. The arbitration priority is always round-robin.

The main subsystem has PCR interconnect that manages the accesses to the peripheral registers and peripheral memories, and provides a global reset for all peripherals. It also supports the capability to selectively enable or disable the clock for each peripheral individually. The PCR also manages the accesses to the system module registers required to configure the device clocks, interrupts, and so forth.

The system module registers include status flags for indicating exception conditions – resets, aborts, errors, and interrupts.

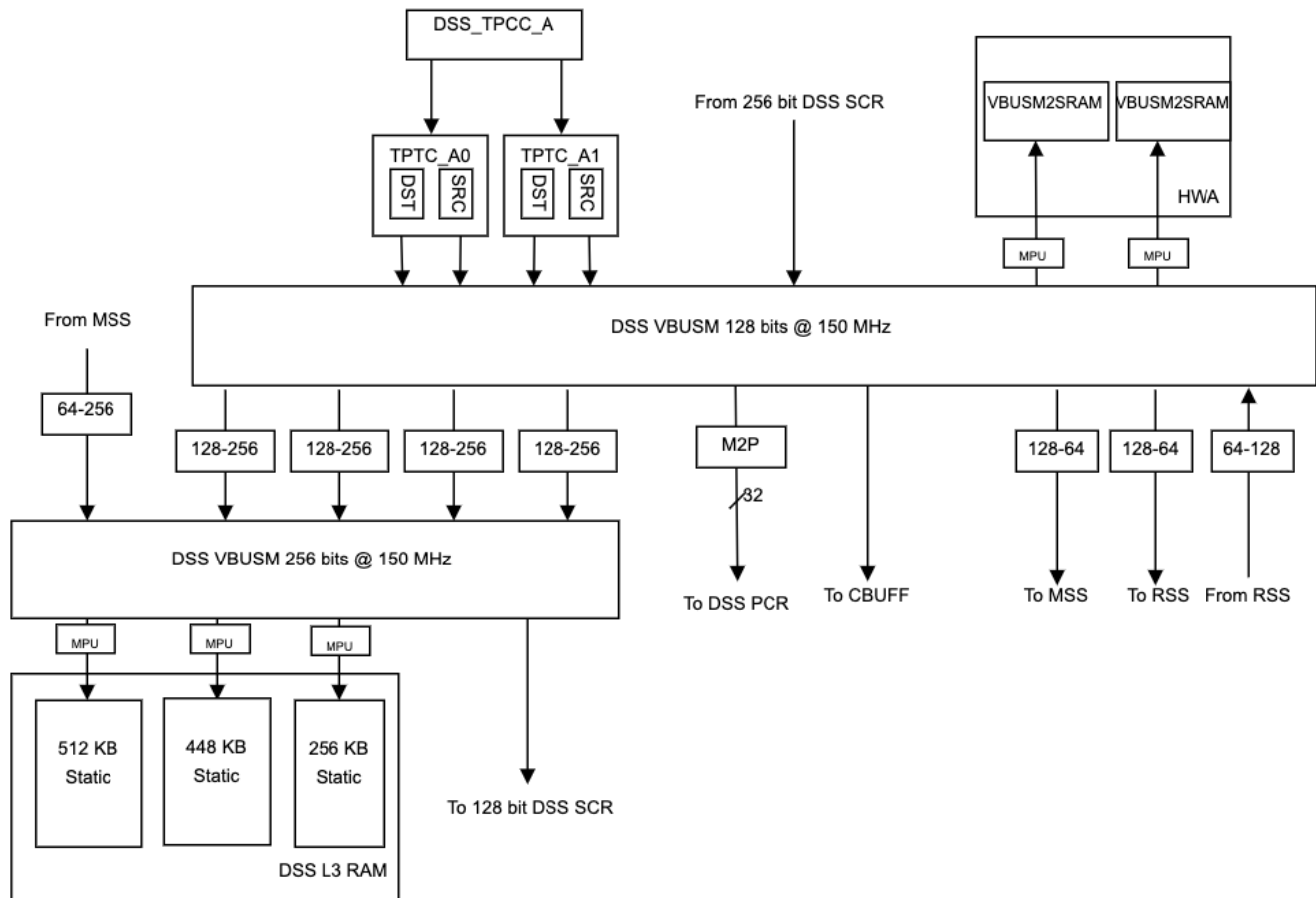


**Figure 3-1. Main Subsystem R5F Infrastructure**



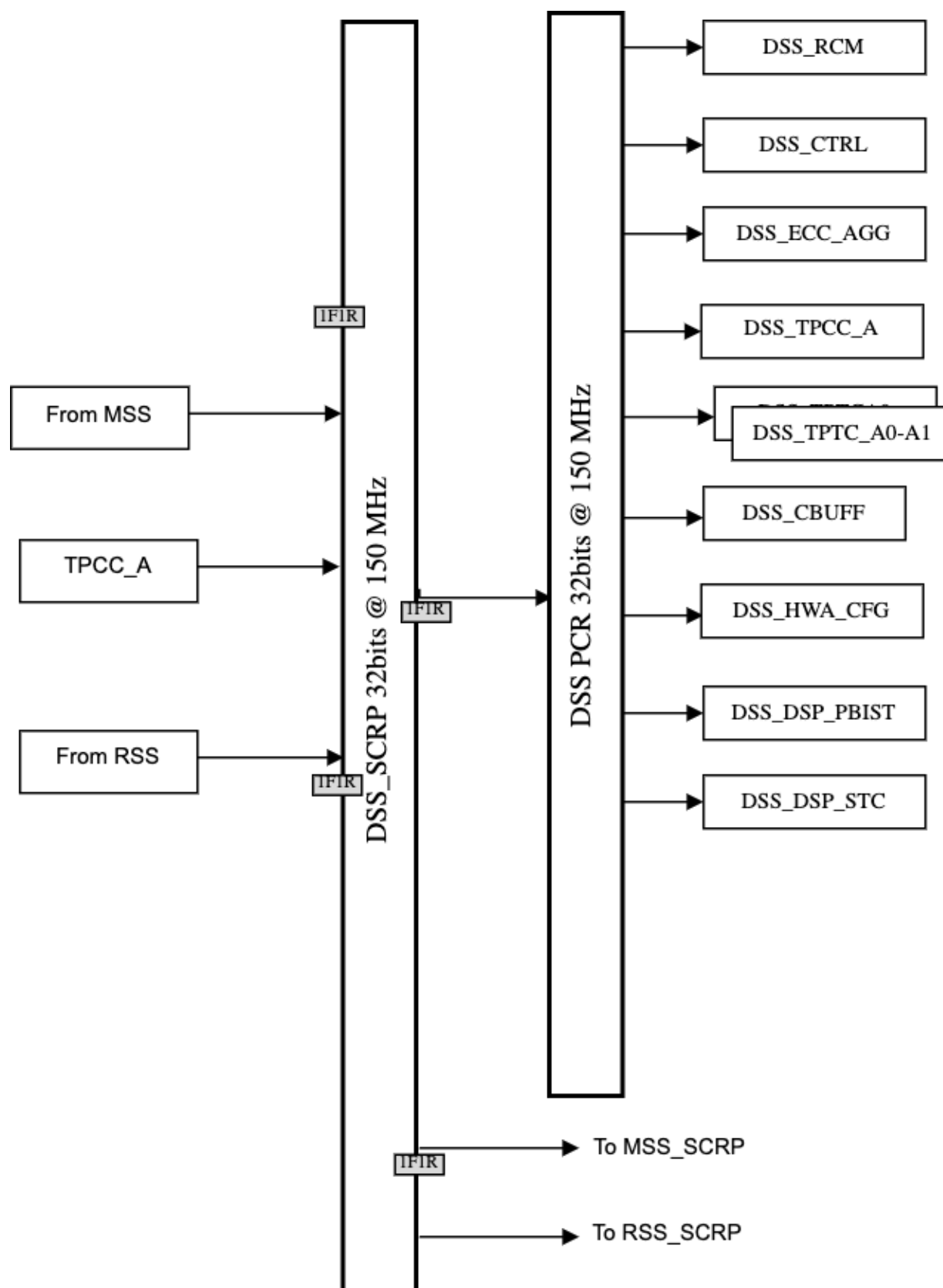
**Figure 3-2. MSS Peripheral Infrastructure**

## 3.2 DSP Subsystem C66x Infrastructure



**Figure 3-3. DSP Subsystem Infrastructure**

The DSS has a hybrid infrastructure of 256-bit and 128-bit data SCRs. This is to match the L3 and DSP MDMA port data widths. The area overhead of hooking up all the other controllers and targets to a 256-bit SCR because the number of bridges required was too large. The MSS Controller also sits on the 256-bit SCR, so that the latency to L3 is reduced.



**Figure 3-4. DSS Peripheral Infrastructure**

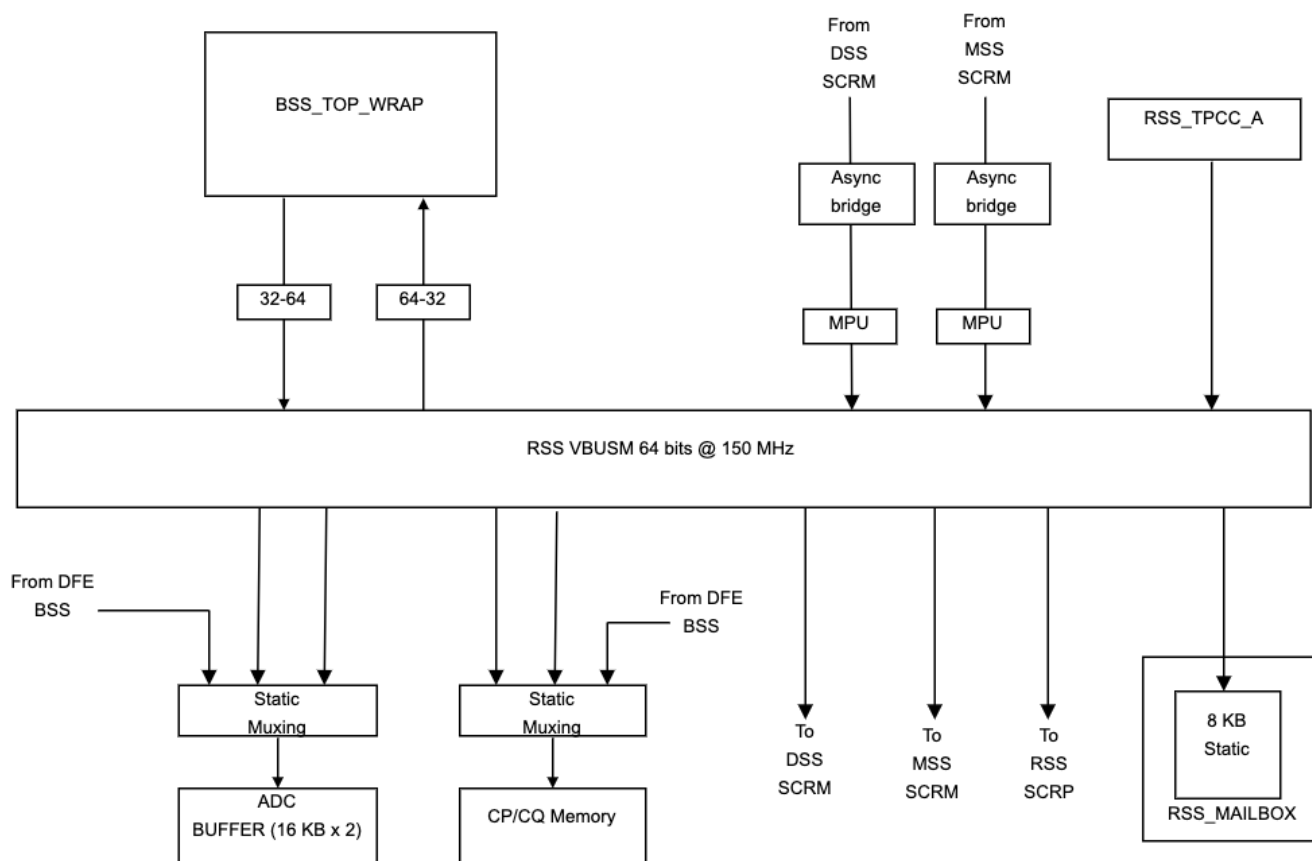


### Note

TPTC-C2, TPTC-C3, TPTC-C4, and TPTC-C5 modules and their functionality are not supported in this family of devices. *Any information regarding these modules has been retained in the documentation solely for the purpose of clarifying memory map read/write attributes. Features noted as “not supported” must not be used.*

## 3.3 Radar Subsystem Infrastructure

The RSS has a 64-bit data SCR



**Figure 3-5. Radar Control Subsystem Infrastructure**

## 3.4 Peripheral Central Resource (PCR)

There are a total of 6 PCRs in the device.

- MSS\_PCR1
- MSS\_PCR2
- DSS\_PCR
- HSM\_PCR
- HSM\_SOC\_PCR
- RSS\_PCR

**Table 3-1. MSS\_PCR1 Mapping to slave**

SI No	Peripheral Name	PCR Region	Quadrants
1	MSS_SPIB_RAM	PCS1	NONE
2	MSS_VIM_R5A	PCS4	NONE

**Table 3-1. MSS\_PCR1 Mapping to slave (continued)**

SI No	Peripheral Name	PCR Region	Quadrants
3	MSS_IOMUX	PCS6	NONE
4	MSS_RCM	PCS8	NONE
5	MSS_CTRL	PCS9	NONE
6	MSS_TOPRCM	PCS10	NONE
7	MSS_PCR1	PS30,31	All Quadrants
8	TOP_PBIST	PS26	0,1
9	MSS_R5SS_STC	PS25	0,1
10	MSS_DCCA	PS24	0
11	MSS_DCCB	PS24	1
12	MSS_DCCC	PS24	2
13	MSS_DCCD	PS24	3
14	MSS_RTIA	PS23	0
15	MSS_RTIB	PS23	1
16	MSS_RTIC	PS23	2
17	MSS_WDT	PS23	3
18	MSS_ESM	PS22	0,1
19	TOP_EFUSE_FARM	PS21	0
20	MSS_CCMR	PS20	0
21	MSS_I2C	PS19	0
22	MSS_GIO	PS18	0,1
23	MSS_ECC_AGG_R5A	PS17	0,1,2,3
24	MSS_ECC_AGG_MSS	PS15	0,1,2,3
25	MSS_SPIA	PS5	0,1
26	MSS_SPIB	PS5	2,3
27	MSS_SCIA	PS4	0
28	MSS_SCIB	PS4	1

**Table 3-2. MSS\_PCR2 Mapping to slave**

SI No	Peripheral Name	PCR Region	Quadrants
1	TOP_AURORA_TX	PCS3	NONE
2	TOP_MDO_INFRA	PCS4	NONE
3	MSS_GPADC_PKT_RAM	PCS6	NONE
4	TOP_CTRL	PCS7	NONE
5	MSS_TPCC_A	PCS8	NONE
6	MSS_TPCC_A0	PCS10	NONE
7	MSS_TPTC_A1	PCS11	NONE
8	MSS_PCR2	PS30,PS31	All Quadrants
9	MSS_ETPWMA	PS28	0

**Table 3-2. MSS\_PCR2 Mapping to slave (continued)**

SI No	Peripheral Name	PCR Region	Quadrants
10	MSS_ETPWMB	PS28	1
11	MSS_ETPWMC	PS28	2
12	MSS_GPADC_REG	PS25	0

**Table 3-3. DSS\_PCR Mapping to slave**

SI No	Peripheral Name	PCR Region	Quadrants
1	DSS_RCM	PCS0	NONE
2	DSS_CTRL	PCS1	NONE
3	DSS_CBUFF	PCS2	NONE
4	DSS_HWA	PCS3	NONE
5	DSS_HWA_EXT	PCS4	NONE
6	DSS_ECC_AGG	PCS5	NONE
7	DSS_TPCC_A	PCS8	NONE
8	DSS_TPTC_A0	PCS11	NONE
9	DSS_TPTC_A1	PCS12	NONE
10	DSS_RESERVED (Unused)	PCS13	NONE
11	DSS_PCR	PS30,PS31	All Quadrants
12	DSS_DSP_PBIST	PS27	0,1

**Table 3-4. RSS\_PCR Mapping to slave**

SI No	Peripheral Name	PCR Region	Quadrants
1	RSS_RCM	PCS0	NONE
2	RSS_CTRL	PCS1	NONE
3	RSS_PROG_FILT	PCS5	NONE
4	MSS2RSS_MPU	PCS6	NONE
5	RSS_TPTC_A	PCS8	NONE
6	RSS_TPTC_A0	PCS11	NONE
7	RSS_ECC_AGG	PCS14	NONE
8	RSS_FILT_COEFF	PCS15	NONE
9	RSS_PROC_CTRL	PCS16	NONE
10	RSS_PROC_ECC_AGG	PCS17	NONE

#### Note

For all the PCR tables above, if any of the PCR region has more than one quadrant then a) if it has quadrant set 0,1 then configure only quadrant 0. b) if it has quadrant set 0,1,2,3 then configure only quadrant 0. c) if it has quadrant set 2,3 then configure only quadrant 2.

### General description about the PCR

The Peripheral Central Resource (PCR) provides decoding and multiplexing between the peripheral bridge and the peripherals and memories. The peripheral bridge is connected to the PCR using the VBUSP. The VBUSP protocol is used to communicate with the peripherals selects (PS), the peripheral memory frame (PCS). Each

PS, PPSE (for system peripherals) frame can be fractioned into 4 quadrants, that can occupy 256 bytes each. For PCS, there is no provision of quadrants.

### Different ways to protect a peripheral using PCR

1. **Using the PMPROT (peripheral memory protection) and PPROT (peripheral protection) registers** - using these registers, it is possible to restrict writes to each one of the non-system peripherals individually to only the privilege mode. Here, PMPROT is being used for PCS region and PPROT registers are being used to protect the PS region.
2. **Using the master-id registers** - using these registers, access to a slave can be restricted for a particular master (maximum 16 masters) using master-id register bits. PCR will respond with an address error (AERROR) for requests from masters which does not have access privilege through master-id registers. To utilize the master-id protection for the peripherals, the Software should first program master-id registers for all generated frames by enabling writes to master-id registers through master-id protect write enable register (MSTIDWRENA). Then enable the master-id check through the master-id enable register (MSTIDENA).

Register size	Number of assigned quadrant	Quadrant 0	Quadrant 1	Quadrant 2	Quadrant 3
<256	1	✓	✓	✓	✓
512> >256	2	✓	-	✓	-
1K> >512	4	✓	-	-	-
>1K	N/A	-	-	-	-

### How the quadrants are being used in a PS frame

The following are the ways in which quadrants are used within a PS frame:

1. The slave uses all the four quadrants: Only the bit corresponding to the quadrant 0 of PSn is implemented. It protects the whole 1K-byte frame. The remaining three bits are not implemented.
2. The slave uses two quadrants: Each quadrant has to be in one of these groups: (Quad 0 and Quad 1) or (Quad 2 and Quad 3). For the group Quad0/Quad1, the bit quadrant 0 protects both quadrants 0 and 1. The bit quadrant 1 is not implemented. For the group Quad2/Quad3, the bit quadrant 2 protects both quadrants 2 and 3. The bit quadrant 3 is not implemented.
3. The slave uses only one quadrant: In this case, the bit, for that specific quadrant will protect the whole peripheral.

### 3.4.1 MSS\_PCR1 Registers

Table 3-5 lists the memory-mapped registers for the MSS\_PCR1 registers. All register offset addresses not listed in Table 3-5 should be considered as reserved locations and the register contents should not be modified.

**Table 3-5. MSS\_PCR1 Registers**

Offset	Acronym	Register Name	Section
0h	PMPROTSET0	PMPROTSET0	<a href="#">Section 3.4.1.1</a>
4h	PMPROTSET1	PMPROTSET1	<a href="#">Section 3.4.1.2</a>
10h	PMPROTCLR0	PMPROTCLR0	<a href="#">Section 3.4.1.3</a>
14h	PMPROTCLR1	PMPROTCLR1	<a href="#">Section 3.4.1.4</a>
20h	PPROTSET_0	PPROTSET_0	<a href="#">Section 3.4.1.5</a>
24h	PPROTSET_1	PPROTSET_1	<a href="#">Section 3.4.1.6</a>
28h	PPROTSET_2	PPROTSET_2	<a href="#">Section 3.4.1.7</a>
2Ch	PPROTSET_3	PPROTSET_3	<a href="#">Section 3.4.1.8</a>
40h	PPROTCLR0	PPROTCLR0	<a href="#">Section 3.4.1.9</a>
44h	PPROTCLR1	PPROTCLR1	<a href="#">Section 3.4.1.10</a>
48h	PPROTCLR2	PPROTCLR2	<a href="#">Section 3.4.1.11</a>
4Ch	PPROTCLR3	PPROTCLR3	<a href="#">Section 3.4.1.12</a>
60h	PCSPWRDWNSET0	PCSPWRDWNSET0	<a href="#">Section 3.4.1.13</a>
64h	PCSPWRDWNSET1	PCSPWRDWNSET1	<a href="#">Section 3.4.1.14</a>
70h	PCSPWRDWNCLR0	PCSPWRDWNCLR0	<a href="#">Section 3.4.1.15</a>
74h	PCSPWRDWNCLR1	PCSPWRDWNCLR1	<a href="#">Section 3.4.1.16</a>
80h	PSPWRDWNSET0	PSPWRDWNSET0	<a href="#">Section 3.4.1.17</a>
84h	PSPWRDWNSET1	PSPWRDWNSET1	<a href="#">Section 3.4.1.18</a>
88h	PSPWRDWNSET2	PSPWRDWNSET2	<a href="#">Section 3.4.1.19</a>
8Ch	PSPWRDWNSET3	PSPWRDWNSET3	<a href="#">Section 3.4.1.20</a>
A0h	PSPWRDWNCLR0	PSPWRDWNCLR0	<a href="#">Section 3.4.1.21</a>
A4h	PSPWRDWNCLR1	PSPWRDWNCLR1	<a href="#">Section 3.4.1.22</a>
A8h	PSPWRDWNCLR2	PSPWRDWNCLR2	<a href="#">Section 3.4.1.23</a>
ACh	PSPWRDWNCLR3	PSPWRDWNCLR3	<a href="#">Section 3.4.1.24</a>
C0h	PDPWRDWNSET	PDPWRDWNSET	<a href="#">Section 3.4.1.25</a>
C4h	PDPWRDWNCLR	PDPWRDWNCLR	<a href="#">Section 3.4.1.26</a>
200h	MSTIDWRENA	MSTIDWRENA	<a href="#">Section 3.4.1.27</a>
204h	MSTIDENA	MSTIDENA	<a href="#">Section 3.4.1.28</a>
208h	MSTIDDIAGCTRL	MSTIDDIAGCTRL	<a href="#">Section 3.4.1.29</a>
300h	PS0MSTID_L	PS0MSTID_L	<a href="#">Section 3.4.1.30</a>
304h	PS0MSTID_H	PS0MSTID_H	<a href="#">Section 3.4.1.31</a>
308h	PS1MSTID_L	PS1MSTID_L	<a href="#">Section 3.4.1.32</a>
30Ch	PS1MSTID_H	PS1MSTID_H	<a href="#">Section 3.4.1.33</a>
310h	PS2MSTID_L	PS2MSTID_L	<a href="#">Section 3.4.1.34</a>
314h	PS2MSTID_H	PS2MSTID_H	<a href="#">Section 3.4.1.35</a>
318h	PS3MSTID_L	PS3MSTID_L	<a href="#">Section 3.4.1.36</a>
31Ch	PS3MSTID_H	PS3MSTID_H	<a href="#">Section 3.4.1.37</a>
320h	PS4MSTID_L	PS4MSTID_L	<a href="#">Section 3.4.1.38</a>
324h	PS4MSTID_H	PS4MSTID_H	<a href="#">Section 3.4.1.39</a>
328h	PS5MSTID_L	PS5MSTID_L	<a href="#">Section 3.4.1.40</a>
32Ch	PS5MSTID_H	PS5MSTID_H	<a href="#">Section 3.4.1.41</a>

**Table 3-5. MSS\_PCR1 Registers (continued)**

Offset	Acronym	Register Name	Section
330h	PS6MSTID_L	PS6MSTID_L	<a href="#">Section 3.4.1.42</a>
334h	PS6MSTID_H	PS6MSTID_H	<a href="#">Section 3.4.1.43</a>
338h	PS7MSTID_L	PS7MSTID_L	<a href="#">Section 3.4.1.44</a>
33Ch	PS7MSTID_H	PS7MSTID_H	<a href="#">Section 3.4.1.45</a>
340h	PS8MSTID_L	PS8MSTID_L	<a href="#">Section 3.4.1.46</a>
344h	PS8MSTID_H	PS8MSTID_H	<a href="#">Section 3.4.1.47</a>
348h	PS9MSTID_L	PS9MSTID_L	<a href="#">Section 3.4.1.48</a>
34Ch	PS9MSTID_H	PS9MSTID_H	<a href="#">Section 3.4.1.49</a>
350h	PS10MSTID_L	PS10MSTID_L	<a href="#">Section 3.4.1.50</a>
354h	PS10MSTID_H	PS10MSTID_H	<a href="#">Section 3.4.1.51</a>
358h	PS11MSTID_L	PS11MSTID_L	<a href="#">Section 3.4.1.52</a>
35Ch	PS11MSTID_H	PS11MSTID_H	<a href="#">Section 3.4.1.53</a>
360h	PS12MSTID_L	PS12MSTID_L	<a href="#">Section 3.4.1.54</a>
364h	PS12MSTID_H	PS12MSTID_H	<a href="#">Section 3.4.1.55</a>
368h	PS13MSTID_L	PS13MSTID_L	<a href="#">Section 3.4.1.56</a>
36Ch	PS13MSTID_H	PS13MSTID_H	<a href="#">Section 3.4.1.57</a>
370h	PS14MSTID_L	PS14MSTID_L	<a href="#">Section 3.4.1.58</a>
374h	PS14MSTID_H	PS14MSTID_H	<a href="#">Section 3.4.1.59</a>
378h	PS15MSTID_L	PS15MSTID_L	<a href="#">Section 3.4.1.60</a>
37Ch	PS15MSTID_H	PS15MSTID_H	<a href="#">Section 3.4.1.61</a>
380h	PS16MSTID_L	PS16MSTID_L	<a href="#">Section 3.4.1.62</a>
384h	PS16MSTID_H	PS16MSTID_H	<a href="#">Section 3.4.1.63</a>
388h	PS17MSTID_L	PS17MSTID_L	<a href="#">Section 3.4.1.64</a>
38Ch	PS17MSTID_H	PS17MSTID_H	<a href="#">Section 3.4.1.65</a>
390h	PS18MSTID_L	PS18MSTID_L	<a href="#">Section 3.4.1.66</a>
394h	PS18MSTID_H	PS18MSTID_H	<a href="#">Section 3.4.1.67</a>
398h	PS19MSTID_L	PS19MSTID_L	<a href="#">Section 3.4.1.68</a>
39Ch	PS19MSTID_H	PS19MSTID_H	<a href="#">Section 3.4.1.69</a>
3A0h	PS20MSTID_L	PS20MSTID_L	<a href="#">Section 3.4.1.70</a>
3A4h	PS20MSTID_H	PS20MSTID_H	<a href="#">Section 3.4.1.71</a>
3A8h	PS21MSTID_L	PS21MSTID_L	<a href="#">Section 3.4.1.72</a>
3ACh	PS21MSTID_H	PS21MSTID_H	<a href="#">Section 3.4.1.73</a>
3B0h	PS22MSTID_L	PS22MSTID_L	<a href="#">Section 3.4.1.74</a>
3B4h	PS22MSTID_H	PS22MSTID_H	<a href="#">Section 3.4.1.75</a>
3B8h	PS23MSTID_L	PS23MSTID_L	<a href="#">Section 3.4.1.76</a>
3BCh	PS23MSTID_H	PS23MSTID_H	<a href="#">Section 3.4.1.77</a>
3C0h	PS24MSTID_L	PS24MSTID_L	<a href="#">Section 3.4.1.78</a>
3C4h	PS24MSTID_H	PS24MSTID_H	<a href="#">Section 3.4.1.79</a>
3C8h	PS25MSTID_L	PS25MSTID_L	<a href="#">Section 3.4.1.80</a>
3CCh	PS25MSTID_H	PS25MSTID_H	<a href="#">Section 3.4.1.81</a>
3D0h	PS26MSTID_L	PS26MSTID_L	<a href="#">Section 3.4.1.82</a>
3D4h	PS26MSTID_H	PS26MSTID_H	<a href="#">Section 3.4.1.83</a>
3D8h	PS27MSTID_L	PS27MSTID_L	<a href="#">Section 3.4.1.84</a>
3DCh	PS27MSTID_H	PS27MSTID_H	<a href="#">Section 3.4.1.85</a>
3E0h	PS28MSTID_L	PS28MSTID_L	<a href="#">Section 3.4.1.86</a>

**Table 3-5. MSS\_PCR1 Registers (continued)**

Offset	Acronym	Register Name	Section
3E4h	PS28MSTID_H	PS28MSTID_H	<a href="#">Section 3.4.1.87</a>
3E8h	PS29MSTID_L	PS29MSTID_L	<a href="#">Section 3.4.1.88</a>
3ECh	PS29MSTID_H	PS29MSTID_H	<a href="#">Section 3.4.1.89</a>
3F0h	PS30MSTID_L	PS30MSTID_L	<a href="#">Section 3.4.1.90</a>
3F4h	PS30MSTID_H	PS30MSTID_H	<a href="#">Section 3.4.1.91</a>
3F8h	PS31MSTID_L	PS31MSTID_L	<a href="#">Section 3.4.1.92</a>
3FCh	PS31MSTID_H	PS31MSTID_H	<a href="#">Section 3.4.1.93</a>
400h	PPS0MSTID_L	PPS0MSTID_L	<a href="#">Section 3.4.1.94</a>
404h	PPS0MSTID_H	PPS0MSTID_H	<a href="#">Section 3.4.1.95</a>
408h	PPS1MSTID_L	PPS1MSTID_L	<a href="#">Section 3.4.1.96</a>
40Ch	PPS1MSTID_H	PPS1MSTID_H	<a href="#">Section 3.4.1.97</a>
410h	PPS2MSTID_L	PPS2MSTID_L	<a href="#">Section 3.4.1.98</a>
414h	PPS2MSTID_H	PPS2MSTID_H	<a href="#">Section 3.4.1.99</a>
418h	PPS3MSTID_L	PPS3MSTID_L	<a href="#">Section 3.4.1.100</a>
41Ch	PPS3MSTID_H	PPS3MSTID_H	<a href="#">Section 3.4.1.101</a>
420h	PPS4MSTID_L	PPS4MSTID_L	<a href="#">Section 3.4.1.102</a>
424h	PPS4MSTID_H	PPS4MSTID_H	<a href="#">Section 3.4.1.103</a>
428h	PPS5MSTID_L	PPS5MSTID_L	<a href="#">Section 3.4.1.104</a>
42Ch	PPS5MSTID_H	PPS5MSTID_H	<a href="#">Section 3.4.1.105</a>
430h	PPS6MSTID_L	PPS6MSTID_L	<a href="#">Section 3.4.1.106</a>
434h	PPS6MSTID_H	PPS6MSTID_H	<a href="#">Section 3.4.1.107</a>
438h	PPS7MSTID_L	PPS7MSTID_L	<a href="#">Section 3.4.1.108</a>
43Ch	PPS7MSTID_H	PPS7MSTID_H	<a href="#">Section 3.4.1.109</a>
440h	PPSE0MSTID_L	PPSE0MSTID_L	<a href="#">Section 3.4.1.110</a>
444h	PPSE0MSTID_H	PPSE0MSTID_H	<a href="#">Section 3.4.1.111</a>
448h	PPSE1MSTID_L	PPSE1MSTID_L	<a href="#">Section 3.4.1.112</a>
44Ch	PPSE1MSTID_H	PPSE1MSTID_H	<a href="#">Section 3.4.1.113</a>
450h	PPSE2MSTID_L	PPSE2MSTID_L	<a href="#">Section 3.4.1.114</a>
454h	PPSE2MSTID_H	PPSE2MSTID_H	<a href="#">Section 3.4.1.115</a>
458h	PPSE3MSTID_L	PPSE3MSTID_L	<a href="#">Section 3.4.1.116</a>
45Ch	PPSE3MSTID_H	PPSE3MSTID_H	<a href="#">Section 3.4.1.117</a>
460h	PPSE4MSTID_L	PPSE4MSTID_L	<a href="#">Section 3.4.1.118</a>
464h	PPSE4MSTID_H	PPSE4MSTID_H	<a href="#">Section 3.4.1.119</a>
468h	PPSE5MSTID_L	PPSE5MSTID_L	<a href="#">Section 3.4.1.120</a>
46Ch	PPSE5MSTID_H	PPSE5MSTID_H	<a href="#">Section 3.4.1.121</a>
470h	PPSE6MSTID_L	PPSE6MSTID_L	<a href="#">Section 3.4.1.122</a>
474h	PPSE6MSTID_H	PPSE6MSTID_H	<a href="#">Section 3.4.1.123</a>
478h	PPSE7MSTID_L	PPSE7MSTID_L	<a href="#">Section 3.4.1.124</a>
47Ch	PPSE7MSTID_H	PPSE7MSTID_H	<a href="#">Section 3.4.1.125</a>
480h	PPSE8MSTID_L	PPSE8MSTID_L	<a href="#">Section 3.4.1.126</a>
484h	PPSE8MSTID_H	PPSE8MSTID_H	<a href="#">Section 3.4.1.127</a>
488h	PPSE9MSTID_L	PPSE9MSTID_L	<a href="#">Section 3.4.1.128</a>
48Ch	PPSE9MSTID_H	PPSE9MSTID_H	<a href="#">Section 3.4.1.129</a>
490h	PPSE10MSTID_L	PPSE10MSTID_L	<a href="#">Section 3.4.1.130</a>
494h	PPSE10MSTID_H	PPSE10MSTID_H	<a href="#">Section 3.4.1.131</a>

**Table 3-5. MSS\_PCR1 Registers (continued)**

Offset	Acronym	Register Name	Section
498h	PPSE11MSTID_L	PPSE11MSTID_L	<a href="#">Section 3.4.1.132</a>
49Ch	PPSE11MSTID_H	PPSE11MSTID_H	<a href="#">Section 3.4.1.133</a>
4A0h	PPSE12MSTID_L	PPSE12MSTID_L	<a href="#">Section 3.4.1.134</a>
4A4h	PPSE12MSTID_H	PPSE12MSTID_H	<a href="#">Section 3.4.1.135</a>
4A8h	PPSE13MSTID_L	PPSE13MSTID_L	<a href="#">Section 3.4.1.136</a>
4ACh	PPSE13MSTID_H	PPSE13MSTID_H	<a href="#">Section 3.4.1.137</a>
4B0h	PPSE14MSTID_L	PPSE14MSTID_L	<a href="#">Section 3.4.1.138</a>
4B4h	PPSE14MSTID_H	PPSE14MSTID_H	<a href="#">Section 3.4.1.139</a>
4B8h	PPSE15MSTID_L	PPSE15MSTID_L	<a href="#">Section 3.4.1.140</a>
4BCh	PPSE15MSTID_H	PPSE15MSTID_H	<a href="#">Section 3.4.1.141</a>
4C0h	PPSE16MSTID_L	PPSE16MSTID_L	<a href="#">Section 3.4.1.142</a>
4C4h	PPSE16MSTID_H	PPSE16MSTID_H	<a href="#">Section 3.4.1.143</a>
4C8h	PPSE17MSTID_L	PPSE17MSTID_L	<a href="#">Section 3.4.1.144</a>
4CCh	PPSE17MSTID_H	PPSE17MSTID_H	<a href="#">Section 3.4.1.145</a>
4D0h	PPSE18MSTID_L	PPSE18MSTID_L	<a href="#">Section 3.4.1.146</a>
4D4h	PPSE18MSTID_H	PPSE18MSTID_H	<a href="#">Section 3.4.1.147</a>
4D8h	PPSE19MSTID_L	PPSE19MSTID_L	<a href="#">Section 3.4.1.148</a>
4DCh	PPSE19MSTID_H	PPSE19MSTID_H	<a href="#">Section 3.4.1.149</a>
4E0h	PPSE20MSTID_L	PPSE20MSTID_L	<a href="#">Section 3.4.1.150</a>
4E4h	PPSE20MSTID_H	PPSE20MSTID_H	<a href="#">Section 3.4.1.151</a>
4E8h	PPSE21MSTID_L	PPSE21MSTID_L	<a href="#">Section 3.4.1.152</a>
4ECh	PPSE21MSTID_H	PPSE21MSTID_H	<a href="#">Section 3.4.1.153</a>
4F0h	PPSE22MSTID_L	PPSE22MSTID_L	<a href="#">Section 3.4.1.154</a>
4F4h	PPSE22MSTID_H	PPSE22MSTID_H	<a href="#">Section 3.4.1.155</a>
4F8h	PPSE23MSTID_L	PPSE23MSTID_L	<a href="#">Section 3.4.1.156</a>
4FCh	PPSE23MSTID_H	PPSE23MSTID_H	<a href="#">Section 3.4.1.157</a>
500h	PPSE24MSTID_L	PPSE24MSTID_L	<a href="#">Section 3.4.1.158</a>
504h	PPSE24MSTID_H	PPSE24MSTID_H	<a href="#">Section 3.4.1.159</a>
508h	PPSE25MSTID_L	PPSE25MSTID_L	<a href="#">Section 3.4.1.160</a>
50Ch	PPSE25MSTID_H	PPSE25MSTID_H	<a href="#">Section 3.4.1.161</a>
510h	PPSE26MSTID_L	PPSE26MSTID_L	<a href="#">Section 3.4.1.162</a>
514h	PPSE26MSTID_H	PPSE26MSTID_H	<a href="#">Section 3.4.1.163</a>
518h	PPSE27MSTID_L	PPSE27MSTID_L	<a href="#">Section 3.4.1.164</a>
51Ch	PPSE27MSTID_H	PPSE27MSTID_H	<a href="#">Section 3.4.1.165</a>
520h	PPSE28MSTID_L	PPSE28MSTID_L	<a href="#">Section 3.4.1.166</a>
524h	PPSE28MSTID_H	PPSE28MSTID_H	<a href="#">Section 3.4.1.167</a>
528h	PPSE29MSTID_L	PPSE29MSTID_L	<a href="#">Section 3.4.1.168</a>
52Ch	PPSE29MSTID_H	PPSE29MSTID_H	<a href="#">Section 3.4.1.169</a>
530h	PPSE30MSTID_L	PPSE30MSTID_L	<a href="#">Section 3.4.1.170</a>
534h	PPSE30MSTID_H	PPSE30MSTID_H	<a href="#">Section 3.4.1.171</a>
538h	PPSE31MSTID_L	PPSE31MSTID_L	<a href="#">Section 3.4.1.172</a>
53Ch	PPSE31MSTID_H	PPSE31MSTID_H	<a href="#">Section 3.4.1.173</a>
540h	PCS0MSTID	PCS0MSTID	<a href="#">Section 3.4.1.174</a>
544h	PCS1MSTID	PCS1MSTID	<a href="#">Section 3.4.1.175</a>
548h	PCS2MSTID	PCS2MSTID	<a href="#">Section 3.4.1.176</a>



**Table 3-5. MSS\_PCR1 Registers (continued)**

Offset	Acronym	Register Name	Section
54Ch	PCS3MSTID	PCS3MSTID	<a href="#">Section 3.4.1.177</a>
550h	PCS4MSTID	PCS4MSTID	<a href="#">Section 3.4.1.178</a>
554h	PCS5MSTID	PCS5MSTID	<a href="#">Section 3.4.1.179</a>
558h	PCS6MSTID	PCS6MSTID	<a href="#">Section 3.4.1.180</a>
55Ch	PCS7MSTID	PCS7MSTID	<a href="#">Section 3.4.1.181</a>
560h	PCS8MSTID	PCS8MSTID	<a href="#">Section 3.4.1.182</a>
564h	PCS9MSTID	PCS9MSTID	<a href="#">Section 3.4.1.183</a>
568h	PCS10MSTID	PCS10MSTID	<a href="#">Section 3.4.1.184</a>
56Ch	PCS11MSTID	PCS11MSTID	<a href="#">Section 3.4.1.185</a>
570h	PCS12MSTID	PCS12MSTID	<a href="#">Section 3.4.1.186</a>
574h	PCS13MSTID	PCS13MSTID	<a href="#">Section 3.4.1.187</a>
578h	PCS14MSTID	PCS14MSTID	<a href="#">Section 3.4.1.188</a>
57Ch	PCS15MSTID	PCS15MSTID	<a href="#">Section 3.4.1.189</a>
580h	PCS16MSTID	PCS16MSTID	<a href="#">Section 3.4.1.190</a>
584h	PCS17MSTID	PCS17MSTID	<a href="#">Section 3.4.1.191</a>
588h	PCS18MSTID	PCS18MSTID	<a href="#">Section 3.4.1.192</a>
58Ch	PCS19MSTID	PCS19MSTID	<a href="#">Section 3.4.1.193</a>
590h	PCS20MSTID	PCS20MSTID	<a href="#">Section 3.4.1.194</a>
594h	PCS21MSTID	PCS21MSTID	<a href="#">Section 3.4.1.195</a>
598h	PCS22MSTID	PCS22MSTID	<a href="#">Section 3.4.1.196</a>
59Ch	PCS23MSTID	PCS23MSTID	<a href="#">Section 3.4.1.197</a>
5A0h	PCS24MSTID	PCS24MSTID	<a href="#">Section 3.4.1.198</a>
5A4h	PCS25MSTID	PCS25MSTID	<a href="#">Section 3.4.1.199</a>
5A8h	PCS26MSTID	PCS26MSTID	<a href="#">Section 3.4.1.200</a>
5ACh	PCS27MSTID	PCS27MSTID	<a href="#">Section 3.4.1.201</a>
5B0h	PCS28MSTID	PCS28MSTID	<a href="#">Section 3.4.1.202</a>
5B4h	PCS29MSTID	PCS29MSTID	<a href="#">Section 3.4.1.203</a>
5B8h	PCS30MSTID	PCS30MSTID	<a href="#">Section 3.4.1.204</a>
5BCh	PCS31MSTID	PCS31MSTID	<a href="#">Section 3.4.1.205</a>
5C0h	PPCS0MSTID	PPCS0MSTID	<a href="#">Section 3.4.1.206</a>
5C4h	PPCS1MSTID	PPCS1MSTID	<a href="#">Section 3.4.1.207</a>
5C8h	PPCS2MSTID	PPCS2MSTID	<a href="#">Section 3.4.1.208</a>
5CCh	PPCS3MSTID	PPCS3MSTID	<a href="#">Section 3.4.1.209</a>
5D0h	PPCS4MSTID	PPCS4MSTID	<a href="#">Section 3.4.1.210</a>
5D4h	PPCS5MSTID	PPCS5MSTID	<a href="#">Section 3.4.1.211</a>
5D8h	PPCS6MSTID	PPCS6MSTID	<a href="#">Section 3.4.1.212</a>
5DCh	PPCS7MSTID	PPCS7MSTID	<a href="#">Section 3.4.1.213</a>
5E0h	PCREXTMSTID	PCREXTMSTID	<a href="#">Section 3.4.1.214</a>

Complex bit access types are encoded to fit into small table cells. [Table 3-6](#) shows the codes that are used for access types in this section.

**Table 3-6. MSS\_PCR1 Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read

**Table 3-6. MSS\_PCR1 Access Type Codes  
(continued)**

Access Type	Code	Description
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 3.4.1.1 PMPROTSET0 Register (Offset = 0h) [Reset = 00000000h]

PMPROTSET0 is shown in [Table 3-7](#).

Return to the [Summary Table](#).

Set-only register to protect PCS frames 0 to 31

**Table 3-7. PMPROTSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
30	PCS30_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
29	PCS29_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
28	PCS28_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
27	PCS27_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-7. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
26	PCS26_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
25	PCS25_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
24	PCS24_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
23	PCS23_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
22	PCS22_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
21	PCS21_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-7. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
20	PCS20_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
19	PCS19_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
18	PCS18_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
17	PCS17_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
16	PCS16_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
15	PCS15_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-7. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	PCS14_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
13	PCS13_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
12	PCS12_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
11	PCS11_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
10	PCS10_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
9	PCS9_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-7. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS8_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
7	PCS7_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
6	PCS6_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
5	PCS5_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
4	PCS4_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
3	PCS3_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.



**Table 3-7. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
1	PCS1_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
0	PCS0_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.2 PMPROTSET1 Register (Offset = 4h) [Reset = 00000000h]

PMPROTSET1 is shown in [Table 3-8](#).

Return to the [Summary Table](#).

Set-only register to protect PCS frames 32 to 63

**Table 3-8. PMPROTSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
30	PCS62_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
29	PCS61_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
28	PCS60_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
27	PCS59_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
26	PCS58_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
25	PCS57_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-8. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS56_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
23	PCS55_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
22	PCS54_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
21	PCS53_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
20	PCS52_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
19	PCS51_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
18	PCS50_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
17	PCS49_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-8. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS48_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
15	PCS47_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
14	PCS46_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
13	PCS45_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
12	PCS44_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
11	PCS43_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
10	PCS42_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
9	PCS41_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-8. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS40_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
7	PCS39_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
6	PCS38_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
5	PCS37_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
4	PCS36_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
3	PCS35_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
2	PCS34_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
1	PCS33_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-8. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS32_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

### 3.4.1.3 PMPROTCLR0 Register (Offset = 10h) [Reset = 00000000h]

PMPROTCLR0 is shown in [Table 3-9](#).

Return to the [Summary Table](#).

Clear-only register to protect PCS frames 0 to 31

**Table 3-9. PMPROTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
30	PCS30_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
29	PCS29_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
28	PCS28_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
27	PCS27_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
26	PCS26_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
25	PCS25_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

**Table 3-9. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS24_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
23	PCS23_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
22	PCS22_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
21	PCS21_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
20	PCS20_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
19	PCS19_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
18	PCS18_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
17	PCS17_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect



**Table 3-9. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS16_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
15	PCS15_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
14	PCS14_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
13	PCS13_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
12	PCS12_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
11	PCS11_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
10	PCS10_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
9	PCS9_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect



**Table 3-9. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS8_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
7	PCS7_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
6	PCS6_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
5	PCS5_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
4	PCS4_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
3	PCS3_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
2	PCS2_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
1	PCS1_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

**Table 3-9. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS0_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

### 3.4.1.4 PMPROTCLR1 Register (Offset = 14h) [Reset = 00000000h]

PMPROTCLR1 is shown in [Table 3-10](#).

Return to the [Summary Table](#).

Clear-only register to protect PCS frames 32 to 63

**Table 3-10. PMPROTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
30	PCS62_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
29	PCS61_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
28	PCS60_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
27	PCS59_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
26	PCS58_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
25	PCS57_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-10. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS56_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
23	PCS55_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
22	PCS54_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
21	PCS53_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
20	PCS52_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
19	PCS51_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
18	PCS50_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
17	PCS49_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-10. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS48_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
15	PCS47_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
14	PCS46_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
13	PCS45_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
12	PCS44_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
11	PCS43_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
10	PCS42_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
9	PCS41_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-10. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS40_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
7	PCS39_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
6	PCS38_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
5	PCS37_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
4	PCS36_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
3	PCS35_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
2	PCS34_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
1	PCS33_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-10. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS32_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect



### 3.4.1.5 PPROTSET\_0 Register (Offset = 20h) [Reset = 00000000h]

PPROTSET\_0 is shown in [Table 3-11](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS0 to PS7

**Table 3-11. PPROTSET\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



**Table 3-11. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-11. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-11. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-11. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

### 3.4.1.6 PPROTSET\_1 Register (Offset = 24h) [Reset = 00000000h]

PPROTSET\_1 is shown in [Table 3-12](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS8 to PS15

**Table 3-12. PPROTSET\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-12. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-12. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect



**Table 3-12. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect



**Table 3-12. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

### 3.4.1.7 PPROTSET\_2 Register (Offset = 28h) [Reset = 00000000h]

PPROTSET\_2 is shown in [Table 3-13](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS16 to PS23

**Table 3-13. PPROTSET\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-13. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-13. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-13. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-13. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

### 3.4.1.8 PPROTSET\_3 Register (Offset = 2Ch) [Reset = 00000000h]

PPROTSET\_3 is shown in [Table 3-14](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS24 to PS31

**Table 3-14. PPROTSET\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



**Table 3-14. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



**Table 3-14. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-14. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-14. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

### 3.4.1.9 PPROTCLR0 Register (Offset = 40h) [Reset = 00000000h]

PPROTCLR0 is shown in [Table 3-15](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS0 to PS7

**Table 3-15. PPROTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-15. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-15. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-15. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



**Table 3-15. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



### 3.4.1.10 PPROTCLR1 Register (Offset = 44h) [Reset = 00000000h]

PPROTCLR1 is shown in [Table 3-16](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS8 to PS15

**Table 3-16. PPROTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-16. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-16. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-16. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-16. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

### 3.4.1.11 PPROTCLR2 Register (Offset = 48h) [Reset = 00000000h]

PPROTCLR2 is shown in [Table 3-17](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS16 to PS23

**Table 3-17. PPROTCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-17. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect



**Table 3-17. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect



**Table 3-17. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-17. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

### 3.4.1.12 PPROTCLR3 Register (Offset = 4Ch) [Reset = 00000000h]

PPROTCLR3 is shown in [Table 3-18](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS24 to PS31

**Table 3-18. PPROTCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-18. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-18. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-18. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-18. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



### 3.4.1.13 PCSPWRDWNSET0 Register (Offset = 60h) [Reset = 00000000h]

PCSPWRDWNSET0 is shown in [Table 3-19](#).

Return to the [Summary Table](#).

Set-only register to powerdown independent (non-shared) PCS frames 0 to 31

**Table 3-19. PCSPWRDWNSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-19. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS22_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-19. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS12_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-19. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.1.14 PCSPWRDWNSET1 Register (Offset = 64h) [Reset = 00000000h]

PCSPWRDWNSET1 is shown in [Table 3-20](#).

Return to the [Summary Table](#).

Set-only register to powerdown independent (non-shared) PCS frames 32 to 63

**Table 3-20. PCSPWRDWNSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-20. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS54_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-20. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS44_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-20. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS34_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect



### 3.4.1.15 PCSPWRDWNCLR0 Register (Offset = 70h) [Reset = 00000000h]

PCSPWRDWNCLR0 is shown in [Table 3-21](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 0 to 31

**Table 3-21. PCSPWRDWNCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-21. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS22_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-21. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS12_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-21. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.1.16 PCSPWRDWNCLR1 Register (Offset = 74h) [Reset = 00000000h]

PCSPWRDWNCLR1 is shown in [Table 3-22](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 32 to 63

**Table 3-22. PCSPWRDWNCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-22. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS54_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-22. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS44_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-22. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS34_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect



### 3.4.1.17 PSPWRDWNSET0 Register (Offset = 80h) [Reset = 00000000h]

PSPWRDWNSET0 is shown in [Table 3-23](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS0 to PS7

**Table 3-23. PSPWRDWNSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-23. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-23. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable bit only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-23. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-23. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.1.18 PSPWRDWNSET1 Register (Offset = 84h) [Reset = 00000000h]

PSPWRDWNSET1 is shown in [Table 3-24](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS8 to PS15

**Table 3-24. PSPWRDWNSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-24. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-24. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect



**Table 3-24. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect



**Table 3-24. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

### 3.4.1.19 PSPWRDWNSET2 Register (Offset = 88h) [Reset = 00000000h]

PSPWRDWNSET2 is shown in [Table 3-25](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS16 to PS23

**Table 3-25. PSPWRDWNSET2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-25. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-25. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-25. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-25. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

### 3.4.1.20 PSPWRDWNSET3 Register (Offset = 8Ch) [Reset = 00000000h]

PSPWRDWNSET3 is shown in [Table 3-26](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS24 to PS31

**Table 3-26. PSPWRDWNSET3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-26. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect



**Table 3-26. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-26. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-26. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

### 3.4.1.21 PSPWRDWNCLR0 Register (Offset = A0h) [Reset = 00000000h]

PSPWRDWNCLR0 is shown in [Table 3-27](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS0 to PS7

**Table 3-27. PSPWRDWNCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-27. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-27. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-27. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-27. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect



### 3.4.1.22 PSPWRDWNCLR1 Register (Offset = A4h) [Reset = 00000000h]

PSPWRDWNCLR1 is shown in [Table 3-28](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS8 to PS15

**Table 3-28. PSPWRDWNCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-28. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-28. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-28. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-28. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

### 3.4.1.23 PSPWRDWNCLR2 Register (Offset = A8h) [Reset = 00000000h]

PSPWRDWNCLR2 is shown in [Table 3-29](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS16 to PS23

**Table 3-29. PSPWRDWNCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-29. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect



**Table 3-29. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect



**Table 3-29. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-29. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

### 3.4.1.24 PSPWRDWNCLR3 Register (Offset = ACh) [Reset = 00000000h]

PSPWRDWNCLR3 is shown in [Table 3-30](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS24 to PS31

**Table 3-30. PSPWRDWNCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_CLR	R/W	0h	
30	PS31_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-30. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
23	PS29_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-30. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
15	PS27_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-30. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7	PS25_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

### 3.4.1.25 PDPWRDWNSET Register (Offset = C0h) [Reset = 00000000h]

PDPWRDWNSET is shown in [Table 3-31](#).

Return to the [Summary Table](#).

Set-only register to powerdown the debug frame

**Table 3-31. PDPWRDWNSET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = Clock to the debug frame needs to be powered down. 0 = Clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get set in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect

### 3.4.1.26 PDPWRDWNCLR Register (Offset = C4h) [Reset = 00000000h]

PDPWRDWNCLR is shown in [Table 3-32](#).

Return to the [Summary Table](#).

Clear-only register to deassert the debug frame's powerdown bit

**Table 3-32. PDPWRDWNCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the debug frame needs to be powered down. 0 = The clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get cleared in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect



### 3.4.1.27 MSTIDWRENA Register (Offset = 200h) [Reset = 00000000h]

MSTIDWRENA is shown in [Table 3-33](#).

Return to the [Summary Table](#).

MasterID Protection Write Enable Register

**Table 3-33. MSTIDWRENA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTIDREG_WRENA	R/W	0h	Readable in both user and privileged modes. 1010 = All master-id registers are unlocked and available for write. others = Writes to all master-id registers are locked. Writable only in privileged mode 1010 = Writes to master-id registers are unlocked. others = Writes to master-id registers are locked.

### 3.4.1.28 MSTIDENA Register (Offset = 204h) [Reset = 00000000h]

MSTIDENA is shown in [Table 3-34](#).

Return to the [Summary Table](#).

MasterID Protection Enable Register

**Table 3-34. MSTIDENA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTID_CHK_EN	R/W	0h	Readable in both user and privileged modes. Writable only in privileged mode 1010 = Enable the master-id feature check. others = Master-id check is disabled.

### 3.4.1.29 MSTIDDIAGCTRL Register (Offset = 208h) [Reset = 00000000h]

MSTIDDIAGCTRL is shown in [Table 3-35](#).

Return to the [Summary Table](#).

MasterID Diagnostic Control Register

**Table 3-35. MSTIDDIAGCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-8	DIAG_CMP_VALUE	R/W	0h	MasterID diagnostic mode control register bits; 4-bit data which is compared with the master-id register of all defined frames during diagnostic mode. Any error in compare logic is indicated through AERROR output from PCR. Readable in both user and privileged modes. Reads the programmed value in diagnostic compare value field. Writable only in privileged mode
7-4	RESERVED	R	0h	Reserved
3-0	DIAG_MODE_EN	R/W	0h	MasterID compare logic diagnostic mode enable bits; 4-bit key for enabling the master-id registers compare logic. Readable in both user and privileged modes. Writable only in privileged mode 1010 = Master-id compare diagnostic mode is enabled. others = Master-id compare diagnostic mode is disabled.

### 3.4.1.30 PS0MSTID\_L Register (Offset = 300h) [Reset = 00000000h]

PS0MSTID\_L is shown in [Table 3-36](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register0\_L

**Table 3-36. PS0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.31 PS0MSTID\_H Register (Offset = 304h) [Reset = 00000000h]

PS0MSTID\_H is shown in [Table 3-37](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register0\_H

**Table 3-37. PS0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.32 PS1MSTID\_L Register (Offset = 308h) [Reset = 00000000h]

PS1MSTID\_L is shown in [Table 3-38](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register1\_L

**Table 3-38. PS1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.33 PS1MSTID\_H Register (Offset = 30Ch) [Reset = 00000000h]

PS1MSTID\_H is shown in [Table 3-39](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register1\_H

**Table 3-39. PS1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.34 PS2MSTID\_L Register (Offset = 310h) [Reset = 00000000h]

PS2MSTID\_L is shown in [Table 3-40](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register2\_L

**Table 3-40. PS2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.35 PS2MSTID\_H Register (Offset = 314h) [Reset = 00000000h]

PS2MSTID\_H is shown in [Table 3-41](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register2\_H

**Table 3-41. PS2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.36 PS3MSTID\_L Register (Offset = 318h) [Reset = 00000000h]

PS3MSTID\_L is shown in [Table 3-42](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register3\_L

**Table 3-42. PS3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.37 PS3MSTID\_H Register (Offset = 31Ch) [Reset = 00000000h]

PS3MSTID\_H is shown in [Table 3-43](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register3\_H

**Table 3-43. PS3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.38 PS4MSTID\_L Register (Offset = 320h) [Reset = 00000000h]

PS4MSTID\_L is shown in [Table 3-44](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register4\_L

**Table 3-44. PS4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.39 PS4MSTID\_H Register (Offset = 324h) [Reset = 00000000h]

PS4MSTID\_H is shown in [Table 3-45](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register4\_H

**Table 3-45. PS4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.40 PS5MSTID\_L Register (Offset = 328h) [Reset = 00000000h]

PS5MSTID\_L is shown in [Table 3-46](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register5\_L

**Table 3-46. PS5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.41 PS5MSTID\_H Register (Offset = 32Ch) [Reset = 00000000h]

PS5MSTID\_H is shown in [Table 3-47](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register5\_H

**Table 3-47. PS5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.42 PS6MSTID\_L Register (Offset = 330h) [Reset = 00000000h]

PS6MSTID\_L is shown in [Table 3-48](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register6\_L

**Table 3-48. PS6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.43 PS6MSTID\_H Register (Offset = 334h) [Reset = 00000000h]

PS6MSTID\_H is shown in [Table 3-49](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register6\_H

**Table 3-49. PS6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.44 PS7MSTID\_L Register (Offset = 338h) [Reset = 00000000h]

PS7MSTID\_L is shown in [Table 3-50](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register7\_L

**Table 3-50. PS7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.45 PS7MSTID\_H Register (Offset = 33Ch) [Reset = 00000000h]

PS7MSTID\_H is shown in [Table 3-51](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register7\_H

**Table 3-51. PS7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.46 PS8MSTID\_L Register (Offset = 340h) [Reset = 00000000h]

PS8MSTID\_L is shown in [Table 3-52](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register8\_L

**Table 3-52. PS8MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.47 PS8MSTID\_H Register (Offset = 344h) [Reset = 00000000h]

PS8MSTID\_H is shown in [Table 3-53](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register8\_H

**Table 3-53. PS8MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.48 PS9MSTID\_L Register (Offset = 348h) [Reset = 00000000h]

PS9MSTID\_L is shown in [Table 3-54](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register9\_L

**Table 3-54. PS9MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.49 PS9MSTID\_H Register (Offset = 34Ch) [Reset = 00000000h]

PS9MSTID\_H is shown in [Table 3-55](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register9\_H

**Table 3-55. PS9MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.50 PS10MSTID\_L Register (Offset = 350h) [Reset = 00000000h]

PS10MSTID\_L is shown in [Table 3-56](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register10\_L

**Table 3-56. PS10MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.51 PS10MSTID\_H Register (Offset = 354h) [Reset = 00000000h]

PS10MSTID\_H is shown in [Table 3-57](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register10\_H

**Table 3-57. PS10MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.52 PS11MSTID\_L Register (Offset = 358h) [Reset = 00000000h]

PS11MSTID\_L is shown in [Table 3-58](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register11\_L

**Table 3-58. PS11MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.53 PS11MSTID\_H Register (Offset = 35Ch) [Reset = 00000000h]

PS11MSTID\_H is shown in [Table 3-59](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register11\_H

**Table 3-59. PS11MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.54 PS12MSTID\_L Register (Offset = 360h) [Reset = 00000000h]

PS12MSTID\_L is shown in [Table 3-60](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register12\_L

**Table 3-60. PS12MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.55 PS12MSTID\_H Register (Offset = 364h) [Reset = 00000000h]

PS12MSTID\_H is shown in [Table 3-61](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register12\_H

**Table 3-61. PS12MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.56 PS13MSTID\_L Register (Offset = 368h) [Reset = 00000000h]

PS13MSTID\_L is shown in [Table 3-62](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register13\_L

**Table 3-62. PS13MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.57 PS13MSTID\_H Register (Offset = 36Ch) [Reset = 00000000h]

PS13MSTID\_H is shown in [Table 3-63](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register13\_H

**Table 3-63. PS13MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.58 PS14MSTID\_L Register (Offset = 370h) [Reset = 00000000h]

PS14MSTID\_L is shown in [Table 3-64](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register14\_L

**Table 3-64. PS14MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.59 PS14MSTID\_H Register (Offset = 374h) [Reset = 00000000h]

PS14MSTID\_H is shown in [Table 3-65](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register14\_H

**Table 3-65. PS14MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.60 PS15MSTID\_L Register (Offset = 378h) [Reset = 00000000h]

PS15MSTID\_L is shown in [Table 3-66](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register15\_L

**Table 3-66. PS15MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.61 PS15MSTID\_H Register (Offset = 37Ch) [Reset = 00000000h]

PS15MSTID\_H is shown in [Table 3-67](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register15\_H

**Table 3-67. PS15MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.62 PS16MSTID\_L Register (Offset = 380h) [Reset = 00000000h]

PS16MSTID\_L is shown in [Table 3-68](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register16\_L

**Table 3-68. PS16MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.63 PS16MSTID\_H Register (Offset = 384h) [Reset = 00000000h]

PS16MSTID\_H is shown in [Table 3-69](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register16\_H

**Table 3-69. PS16MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.64 PS17MSTID\_L Register (Offset = 388h) [Reset = 00000000h]

PS17MSTID\_L is shown in [Table 3-70](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register17\_L

**Table 3-70. PS17MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.65 PS17MSTID\_H Register (Offset = 38Ch) [Reset = 00000000h]

PS17MSTID\_H is shown in [Table 3-71](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register17\_H

**Table 3-71. PS17MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.66 PS18MSTID\_L Register (Offset = 390h) [Reset = 00000000h]

PS18MSTID\_L is shown in [Table 3-72](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register18\_L

**Table 3-72. PS18MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.67 PS18MSTID\_H Register (Offset = 394h) [Reset = 00000000h]

PS18MSTID\_H is shown in [Table 3-73](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register18\_H

**Table 3-73. PS18MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.68 PS19MSTID\_L Register (Offset = 398h) [Reset = 00000000h]

PS19MSTID\_L is shown in [Table 3-74](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register19\_L

**Table 3-74. PS19MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.69 PS19MSTID\_H Register (Offset = 39Ch) [Reset = 00000000h]

PS19MSTID\_H is shown in [Table 3-75](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register19\_H

**Table 3-75. PS19MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.70 PS20MSTID\_L Register (Offset = 3A0h) [Reset = 00000000h]

PS20MSTID\_L is shown in [Table 3-76](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register20\_L

**Table 3-76. PS20MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.71 PS20MSTID\_H Register (Offset = 3A4h) [Reset = 00000000h]

PS20MSTID\_H is shown in [Table 3-77](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register20\_H

**Table 3-77. PS20MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.72 PS21MSTID\_L Register (Offset = 3A8h) [Reset = 00000000h]

PS21MSTID\_L is shown in [Table 3-78](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register21\_L

**Table 3-78. PS21MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.73 PS21MSTID\_H Register (Offset = 3ACh) [Reset = 00000000h]

PS21MSTID\_H is shown in [Table 3-79](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register21\_H

**Table 3-79. PS21MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.74 PS22MSTID\_L Register (Offset = 3B0h) [Reset = 00000000h]

PS22MSTID\_L is shown in [Table 3-80](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register22\_L

**Table 3-80. PS22MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.75 PS22MSTID\_H Register (Offset = 3B4h) [Reset = 00000000h]

PS22MSTID\_H is shown in [Table 3-81](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register22\_H

**Table 3-81. PS22MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.76 PS23MSTID\_L Register (Offset = 3B8h) [Reset = 00000000h]

PS23MSTID\_L is shown in [Table 3-82](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register23\_L

**Table 3-82. PS23MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.77 PS23MSTID\_H Register (Offset = 3BCh) [Reset = 00000000h]

PS23MSTID\_H is shown in [Table 3-83](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register23\_H

**Table 3-83. PS23MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.78 PS24MSTID\_L Register (Offset = 3C0h) [Reset = 00000000h]

PS24MSTID\_L is shown in [Table 3-84](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register24\_L

**Table 3-84. PS24MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.79 PS24MSTID\_H Register (Offset = 3C4h) [Reset = 00000000h]

PS24MSTID\_H is shown in [Table 3-85](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register24\_H

**Table 3-85. PS24MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.80 PS25MSTID\_L Register (Offset = 3C8h) [Reset = 00000000h]

PS25MSTID\_L is shown in [Table 3-86](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register25\_L

**Table 3-86. PS25MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.81 PS25MSTID\_H Register (Offset = 3CCh) [Reset = 00000000h]

PS25MSTID\_H is shown in [Table 3-87](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register25\_H

**Table 3-87. PS25MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.82 PS26MSTID\_L Register (Offset = 3D0h) [Reset = 00000000h]

PS26MSTID\_L is shown in [Table 3-88](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register26\_L

**Table 3-88. PS26MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.83 PS26MSTID\_H Register (Offset = 3D4h) [Reset = 00000000h]

PS26MSTID\_H is shown in [Table 3-89](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register26\_H

**Table 3-89. PS26MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.84 PS27MSTID\_L Register (Offset = 3D8h) [Reset = 00000000h]

PS27MSTID\_L is shown in [Table 3-90](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register27\_L

**Table 3-90. PS27MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.85 PS27MSTID\_H Register (Offset = 3DCh) [Reset = 00000000h]

PS27MSTID\_H is shown in [Table 3-91](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register27\_H

**Table 3-91. PS27MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.86 PS28MSTID\_L Register (Offset = 3E0h) [Reset = 00000000h]

PS28MSTID\_L is shown in [Table 3-92](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register28\_L

**Table 3-92. PS28MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.87 PS28MSTID\_H Register (Offset = 3E4h) [Reset = 00000000h]

PS28MSTID\_H is shown in [Table 3-93](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register28\_H

**Table 3-93. PS28MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.88 PS29MSTID\_L Register (Offset = 3E8h) [Reset = 00000000h]

PS29MSTID\_L is shown in [Table 3-94](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register29\_L

**Table 3-94. PS29MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.89 PS29MSTID\_H Register (Offset = 3ECh) [Reset = 00000000h]

PS29MSTID\_H is shown in [Table 3-95](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register29\_H

**Table 3-95. PS29MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.90 PS30MSTID\_L Register (Offset = 3F0h) [Reset = 00000000h]

PS30MSTID\_L is shown in [Table 3-96](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register30\_L

**Table 3-96. PS30MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.91 PS30MSTID\_H Register (Offset = 3F4h) [Reset = 00000000h]

PS30MSTID\_H is shown in [Table 3-97](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register30\_H

**Table 3-97. PS30MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.92 PS31MSTID\_L Register (Offset = 3F8h) [Reset = 00000000h]

PS31MSTID\_L is shown in [Table 3-98](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register31\_L

**Table 3-98. PS31MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.93 PS31MSTID\_H Register (Offset = 3FCh) [Reset = 00000000h]

PS31MSTID\_H is shown in [Table 3-99](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register31\_H

**Table 3-99. PS31MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.94 PPS0MSTID\_L Register (Offset = 400h) [Reset = 00000000h]

PPS0MSTID\_L is shown in [Table 3-100](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register0\_L

**Table 3-100. PPS0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.95 PPS0MSTID\_H Register (Offset = 404h) [Reset = 00000000h]

PPS0MSTID\_H is shown in [Table 3-101](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register0\_H

**Table 3-101. PPS0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.96 PPS1MSTID\_L Register (Offset = 408h) [Reset = 00000000h]

PPS1MSTID\_L is shown in [Table 3-102](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register1\_L

**Table 3-102. PPS1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.97 PPS1MSTID\_H Register (Offset = 40Ch) [Reset = 00000000h]

PPS1MSTID\_H is shown in [Table 3-103](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register1\_H

**Table 3-103. PPS1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.98 PPS2MSTID\_L Register (Offset = 410h) [Reset = 00000000h]

PPS2MSTID\_L is shown in [Table 3-104](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register2\_L

**Table 3-104. PPS2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.99 PPS2MSTID\_H Register (Offset = 414h) [Reset = 00000000h]

PPS2MSTID\_H is shown in [Table 3-105](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register2\_H

**Table 3-105. PPS2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.100 PPS3MSTID\_L Register (Offset = 418h) [Reset = 00000000h]

PPS3MSTID\_L is shown in [Table 3-106](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register3\_L

**Table 3-106. PPS3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.101 PPS3MSTID\_H Register (Offset = 41Ch) [Reset = 00000000h]

PPS3MSTID\_H is shown in [Table 3-107](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register3\_H

**Table 3-107. PPS3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.102 PPS4MSTID\_L Register (Offset = 420h) [Reset = 00000000h]

PPS4MSTID\_L is shown in [Table 3-108](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register4\_L

**Table 3-108. PPS4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.103 PPS4MSTID\_H Register (Offset = 424h) [Reset = 00000000h]

PPS4MSTID\_H is shown in [Table 3-109](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register4\_H

**Table 3-109. PPS4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.104 PPS5MSTID\_L Register (Offset = 428h) [Reset = 00000000h]

PPS5MSTID\_L is shown in [Table 3-110](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register5\_L

**Table 3-110. PPS5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.105 PPS5MSTID\_H Register (Offset = 42Ch) [Reset = 00000000h]

PPS5MSTID\_H is shown in [Table 3-111](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register5\_H

**Table 3-111. PPS5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.106 PPS6MSTID\_L Register (Offset = 430h) [Reset = 00000000h]

PPS6MSTID\_L is shown in [Table 3-112](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register6\_L

**Table 3-112. PPS6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.107 PPS6MSTID\_H Register (Offset = 434h) [Reset = 00000000h]

PPS6MSTID\_H is shown in [Table 3-113](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register6\_H

**Table 3-113. PPS6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.108 PPS7MSTID\_L Register (Offset = 438h) [Reset = 00000000h]

PPS7MSTID\_L is shown in [Table 3-114](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register7\_L

**Table 3-114. PPS7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.109 PPS7MSTID\_H Register (Offset = 43Ch) [Reset = 00000000h]

PPS7MSTID\_H is shown in [Table 3-115](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register7\_H

**Table 3-115. PPS7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.110 PPSE0MSTID\_L Register (Offset = 440h) [Reset = 00000000h]

PPSE0MSTID\_L is shown in [Table 3-116](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register0\_L

**Table 3-116. PPSE0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.111 PPSE0MSTID\_H Register (Offset = 444h) [Reset = 00000000h]

PPSE0MSTID\_H is shown in [Table 3-117](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register0\_H

**Table 3-117. PPSE0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.112 PPSE1MSTID\_L Register (Offset = 448h) [Reset = 00000000h]

PPSE1MSTID\_L is shown in [Table 3-118](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register1\_L

**Table 3-118. PPSE1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.113 PPSE1MSTID\_H Register (Offset = 44Ch) [Reset = 00000000h]

PPSE1MSTID\_H is shown in [Table 3-119](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register1\_H

**Table 3-119. PPSE1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.114 PPSE2MSTID\_L Register (Offset = 450h) [Reset = 00000000h]

PPSE2MSTID\_L is shown in [Table 3-120](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register2\_L

**Table 3-120. PPSE2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.115 PPSE2MSTID\_H Register (Offset = 454h) [Reset = 00000000h]

PPSE2MSTID\_H is shown in [Table 3-121](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register2\_H

**Table 3-121. PPSE2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.116 PPSE3MSTID\_L Register (Offset = 458h) [Reset = 00000000h]

PPSE3MSTID\_L is shown in [Table 3-122](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register3\_L

**Table 3-122. PPSE3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.117 PPSE3MSTID\_H Register (Offset = 45Ch) [Reset = 00000000h]

PPSE3MSTID\_H is shown in [Table 3-123](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register3\_H

**Table 3-123. PPSE3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.118 PPSE4MSTID\_L Register (Offset = 460h) [Reset = 00000000h]

PPSE4MSTID\_L is shown in [Table 3-124](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register4\_L

**Table 3-124. PPSE4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.119 PPSE4MSTID\_H Register (Offset = 464h) [Reset = 00000000h]

PPSE4MSTID\_H is shown in [Table 3-125](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register4\_H

**Table 3-125. PPSE4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.120 PPSE5MSTID\_L Register (Offset = 468h) [Reset = 00000000h]

PPSE5MSTID\_L is shown in [Table 3-126](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register5\_L

**Table 3-126. PPSE5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.121 PPSE5MSTID\_H Register (Offset = 46Ch) [Reset = 00000000h]

PPSE5MSTID\_H is shown in [Table 3-127](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register5\_H

**Table 3-127. PPSE5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.122 PPSE6MSTID\_L Register (Offset = 470h) [Reset = 00000000h]

PPSE6MSTID\_L is shown in [Table 3-128](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register6\_L

**Table 3-128. PPSE6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.123 PPSE6MSTID\_H Register (Offset = 474h) [Reset = 00000000h]

PPSE6MSTID\_H is shown in [Table 3-129](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register6\_H

**Table 3-129. PPSE6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.124 PPSE7MSTID\_L Register (Offset = 478h) [Reset = 00000000h]

PPSE7MSTID\_L is shown in [Table 3-130](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register7\_L

**Table 3-130. PPSE7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.125 PPSE7MSTID\_H Register (Offset = 47Ch) [Reset = 00000000h]

PPSE7MSTID\_H is shown in [Table 3-131](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register7\_H

**Table 3-131. PPSE7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.126 PPSE8MSTID\_L Register (Offset = 480h) [Reset = 00000000h]

PPSE8MSTID\_L is shown in [Table 3-132](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register8\_L

**Table 3-132. PPSE8MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.127 PPSE8MSTID\_H Register (Offset = 484h) [Reset = 00000000h]

PPSE8MSTID\_H is shown in [Table 3-133](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register8\_H

**Table 3-133. PPSE8MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.128 PPSE9MSTID\_L Register (Offset = 488h) [Reset = 00000000h]

PPSE9MSTID\_L is shown in [Table 3-134](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register9\_L

**Table 3-134. PPSE9MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.129 PPSE9MSTID\_H Register (Offset = 48Ch) [Reset = 00000000h]

PPSE9MSTID\_H is shown in [Table 3-135](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register9\_H

**Table 3-135. PPSE9MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.130 PPSE10MSTID\_L Register (Offset = 490h) [Reset = 00000000h]

PPSE10MSTID\_L is shown in [Table 3-136](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register10\_L

**Table 3-136. PPSE10MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.131 PPSE10MSTID\_H Register (Offset = 494h) [Reset = 00000000h]

PPSE10MSTID\_H is shown in [Table 3-137](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register10\_H

**Table 3-137. PPSE10MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.132 PPSE11MSTID\_L Register (Offset = 498h) [Reset = 00000000h]

PPSE11MSTID\_L is shown in [Table 3-138](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register11\_L

**Table 3-138. PPSE11MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.133 PPSE11MSTID\_H Register (Offset = 49Ch) [Reset = 00000000h]

PPSE11MSTID\_H is shown in [Table 3-139](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register11\_H

**Table 3-139. PPSE11MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.134 PPSE12MSTID\_L Register (Offset = 4A0h) [Reset = 00000000h]

PPSE12MSTID\_L is shown in [Table 3-140](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register12\_L

**Table 3-140. PPSE12MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.135 PPSE12MSTID\_H Register (Offset = 4A4h) [Reset = 00000000h]

PPSE12MSTID\_H is shown in [Table 3-141](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register12\_H

**Table 3-141. PPSE12MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.136 PPSE13MSTID\_L Register (Offset = 4A8h) [Reset = 00000000h]

PPSE13MSTID\_L is shown in [Table 3-142](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register13\_L

**Table 3-142. PPSE13MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.137 PPSE13MSTID\_H Register (Offset = 4ACh) [Reset = 00000000h]

PPSE13MSTID\_H is shown in [Table 3-143](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register13\_H

**Table 3-143. PPSE13MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.138 PPSE14MSTID\_L Register (Offset = 4B0h) [Reset = 00000000h]

PPSE14MSTID\_L is shown in [Table 3-144](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register14\_L

**Table 3-144. PPSE14MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.139 PPSE14MSTID\_H Register (Offset = 4B4h) [Reset = 00000000h]

PPSE14MSTID\_H is shown in [Table 3-145](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register14\_H

**Table 3-145. PPSE14MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.140 PPSE15MSTID\_L Register (Offset = 4B8h) [Reset = 00000000h]

PPSE15MSTID\_L is shown in [Table 3-146](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register15\_L

**Table 3-146. PPSE15MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.141 PPSE15MSTID\_H Register (Offset = 4BCh) [Reset = 00000000h]

PPSE15MSTID\_H is shown in [Table 3-147](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register15\_H

**Table 3-147. PPSE15MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.142 PPSE16MSTID\_L Register (Offset = 4C0h) [Reset = 00000000h]

PPSE16MSTID\_L is shown in [Table 3-148](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register16\_L

**Table 3-148. PPSE16MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.143 PPSE16MSTID\_H Register (Offset = 4C4h) [Reset = 00000000h]

PPSE16MSTID\_H is shown in [Table 3-149](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register16\_H

**Table 3-149. PPSE16MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.144 PPSE17MSTID\_L Register (Offset = 4C8h) [Reset = 00000000h]

PPSE17MSTID\_L is shown in [Table 3-150](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register17\_L

**Table 3-150. PPSE17MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.145 PPSE17MSTID\_H Register (Offset = 4CCh) [Reset = 00000000h]

PPSE17MSTID\_H is shown in [Table 3-151](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register17\_H

**Table 3-151. PPSE17MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.146 PPSE18MSTID\_L Register (Offset = 4D0h) [Reset = 00000000h]

PPSE18MSTID\_L is shown in [Table 3-152](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register18\_L

**Table 3-152. PPSE18MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.147 PPSE18MSTID\_H Register (Offset = 4D4h) [Reset = 00000000h]

PPSE18MSTID\_H is shown in [Table 3-153](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register18\_H

**Table 3-153. PPSE18MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.148 PPSE19MSTID\_L Register (Offset = 4D8h) [Reset = 00000000h]

PPSE19MSTID\_L is shown in [Table 3-154](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register19\_L

**Table 3-154. PPSE19MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.149 PPSE19MSTID\_H Register (Offset = 4DCh) [Reset = 00000000h]

PPSE19MSTID\_H is shown in [Table 3-155](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register19\_H

**Table 3-155. PPSE19MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.150 PPSE20MSTID\_L Register (Offset = 4E0h) [Reset = 00000000h]

PPSE20MSTID\_L is shown in [Table 3-156](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register20\_L

**Table 3-156. PPSE20MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.151 PPSE20MSTID\_H Register (Offset = 4E4h) [Reset = 00000000h]

PPSE20MSTID\_H is shown in [Table 3-157](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register20\_H

**Table 3-157. PPSE20MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.152 PPSE21MSTID\_L Register (Offset = 4E8h) [Reset = 00000000h]

PPSE21MSTID\_L is shown in [Table 3-158](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register21\_L

**Table 3-158. PPSE21MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.153 PPSE21MSTID\_H Register (Offset = 4ECh) [Reset = 00000000h]

PPSE21MSTID\_H is shown in [Table 3-159](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register21\_H

**Table 3-159. PPSE21MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.154 PPSE22MSTID\_L Register (Offset = 4F0h) [Reset = 00000000h]

PPSE22MSTID\_L is shown in [Table 3-160](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register22\_L

**Table 3-160. PPSE22MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.155 PPSE22MSTID\_H Register (Offset = 4F4h) [Reset = 00000000h]

PPSE22MSTID\_H is shown in [Table 3-161](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register22\_H

**Table 3-161. PPSE22MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.156 PPSE23MSTID\_L Register (Offset = 4F8h) [Reset = 00000000h]

PPSE23MSTID\_L is shown in [Table 3-162](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register23\_L

**Table 3-162. PPSE23MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.157 PPSE23MSTID\_H Register (Offset = 4FCh) [Reset = 00000000h]

PPSE23MSTID\_H is shown in [Table 3-163](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register23\_H

**Table 3-163. PPSE23MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.158 PPSE24MSTID\_L Register (Offset = 500h) [Reset = 00000000h]

PPSE24MSTID\_L is shown in [Table 3-164](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register24\_L

**Table 3-164. PPSE24MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.159 PPSE24MSTID\_H Register (Offset = 504h) [Reset = 00000000h]

PPSE24MSTID\_H is shown in [Table 3-165](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register24\_H

**Table 3-165. PPSE24MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.160 PPSE25MSTID\_L Register (Offset = 508h) [Reset = 00000000h]

PPSE25MSTID\_L is shown in [Table 3-166](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register25\_L

**Table 3-166. PPSE25MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.161 PPSE25MSTID\_H Register (Offset = 50Ch) [Reset = 00000000h]

PPSE25MSTID\_H is shown in [Table 3-167](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register25\_H

**Table 3-167. PPSE25MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.162 PPSE26MSTID\_L Register (Offset = 510h) [Reset = 00000000h]

PPSE26MSTID\_L is shown in [Table 3-168](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register26\_L

**Table 3-168. PPSE26MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.163 PPSE26MSTID\_H Register (Offset = 514h) [Reset = 00000000h]

PPSE26MSTID\_H is shown in [Table 3-169](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register26\_H

**Table 3-169. PPSE26MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.164 PPSE27MSTID\_L Register (Offset = 518h) [Reset = 00000000h]

PPSE27MSTID\_L is shown in [Table 3-170](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register27\_L

**Table 3-170. PPSE27MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.165 PPSE27MSTID\_H Register (Offset = 51Ch) [Reset = 00000000h]

PPSE27MSTID\_H is shown in [Table 3-171](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register27\_H

**Table 3-171. PPSE27MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.166 PPSE28MSTID\_L Register (Offset = 520h) [Reset = 00000000h]

PPSE28MSTID\_L is shown in [Table 3-172](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register28\_L

**Table 3-172. PPSE28MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.167 PPSE28MSTID\_H Register (Offset = 524h) [Reset = 00000000h]

PPSE28MSTID\_H is shown in [Table 3-173](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register28\_H

**Table 3-173. PPSE28MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.168 PPSE29MSTID\_L Register (Offset = 528h) [Reset = 00000000h]

PPSE29MSTID\_L is shown in [Table 3-174](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register29\_L

**Table 3-174. PPSE29MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.169 PPSE29MSTID\_H Register (Offset = 52Ch) [Reset = 00000000h]

PPSE29MSTID\_H is shown in [Table 3-175](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register29\_H

**Table 3-175. PPSE29MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.170 PPSE30MSTID\_L Register (Offset = 530h) [Reset = 00000000h]

PPSE30MSTID\_L is shown in [Table 3-176](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register30\_L

**Table 3-176. PPSE30MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE30_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.171 PPSE30MSTID\_H Register (Offset = 534h) [Reset = 00000000h]

PPSE30MSTID\_H is shown in [Table 3-177](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register30\_H

**Table 3-177. PPSE30MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.172 PPSE31MSTID\_L Register (Offset = 538h) [Reset = 00000000h]

PPSE31MSTID\_L is shown in [Table 3-178](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register31\_L

**Table 3-178. PPSE31MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.173 PPSE31MSTID\_H Register (Offset = 53Ch) [Reset = 00000000h]

PPSE31MSTID\_H is shown in [Table 3-179](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register31\_H

**Table 3-179. PPSE31MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.174 PCS0MSTID Register (Offset = 540h) [Reset = 00000000h]

PCS0MSTID is shown in [Table 3-180](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register0

**Table 3-180. PCS0MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS1MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS0MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.175 PCS1MSTID Register (Offset = 544h) [Reset = 00000000h]

PCS1MSTID is shown in [Table 3-181](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register1

**Table 3-181. PCS1MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS3MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS2MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.176 PCS2MSTID Register (Offset = 548h) [Reset = 00000000h]

PCS2MSTID is shown in [Table 3-182](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register2

**Table 3-182. PCS2MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS5MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS4MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.177 PCS3MSTID Register (Offset = 54Ch) [Reset = 00000000h]

PCS3MSTID is shown in [Table 3-183](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register3

**Table 3-183. PCS3MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS7MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS6MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.178 PCS4MSTID Register (Offset = 550h) [Reset = 00000000h]

PCS4MSTID is shown in [Table 3-184](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register4

**Table 3-184. PCS4MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS9MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS8MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.1.179 PCS5MSTID Register (Offset = 554h) [Reset = 00000000h]

PCS5MSTID is shown in [Table 3-185](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register5

**Table 3-185. PCS5MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS11MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS10MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.180 PCS6MSTID Register (Offset = 558h) [Reset = 00000000h]

PCS6MSTID is shown in [Table 3-186](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register6

**Table 3-186. PCS6MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS13MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS12MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.181 PCS7MSTID Register (Offset = 55Ch) [Reset = 00000000h]

PCS7MSTID is shown in [Table 3-187](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register7

**Table 3-187. PCS7MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS15MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS14MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.182 PCS8MSTID Register (Offset = 560h) [Reset = 00000000h]

PCS8MSTID is shown in [Table 3-188](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register8

**Table 3-188. PCS8MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS17MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS16MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.183 PCS9MSTID Register (Offset = 564h) [Reset = 00000000h]

PCS9MSTID is shown in [Table 3-189](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register9

**Table 3-189. PCS9MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS19MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS18MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.184 PCS10MSTID Register (Offset = 568h) [Reset = 00000000h]

PCS10MSTID is shown in [Table 3-190](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register10

**Table 3-190. PCS10MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS21MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS20MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.185 PCS11MSTID Register (Offset = 56Ch) [Reset = 00000000h]

PCS11MSTID is shown in [Table 3-191](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register11

**Table 3-191. PCS11MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS23MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS22MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.186 PCS12MSTID Register (Offset = 570h) [Reset = 00000000h]

PCS12MSTID is shown in [Table 3-192](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register12

**Table 3-192. PCS12MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS25MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS24MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.1.187 PCS13MSTID Register (Offset = 574h) [Reset = 00000000h]

PCS13MSTID is shown in [Table 3-193](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register13

**Table 3-193. PCS13MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS27MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS26MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.188 PCS14MSTID Register (Offset = 578h) [Reset = 00000000h]

PCS14MSTID is shown in [Table 3-194](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register14

**Table 3-194. PCS14MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS29MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS28MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.189 PCS15MSTID Register (Offset = 57Ch) [Reset = 00000000h]

PCS15MSTID is shown in [Table 3-195](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register15

**Table 3-195. PCS15MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS31MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS30MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.190 PCS16MSTID Register (Offset = 580h) [Reset = 00000000h]

PCS16MSTID is shown in [Table 3-196](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register16

**Table 3-196. PCS16MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS33MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS32MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.191 PCS17MSTID Register (Offset = 584h) [Reset = 00000000h]

PCS17MSTID is shown in [Table 3-197](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register17

**Table 3-197. PCS17MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS35MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS34MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.192 PCS18MSTID Register (Offset = 588h) [Reset = 00000000h]

PCS18MSTID is shown in [Table 3-198](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register18

**Table 3-198. PCS18MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS37MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS36MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.193 PCS19MSTID Register (Offset = 58Ch) [Reset = 00000000h]

PCS19MSTID is shown in [Table 3-199](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register19

**Table 3-199. PCS19MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS39MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS38MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.194 PCS20MSTID Register (Offset = 590h) [Reset = 00000000h]

PCS20MSTID is shown in [Table 3-200](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register20

**Table 3-200. PCS20MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS41MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS40MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.1.195 PCS21MSTID Register (Offset = 594h) [Reset = 00000000h]

PCS21MSTID is shown in [Table 3-201](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register21

**Table 3-201. PCS21MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS43MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS42MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.196 PCS22MSTID Register (Offset = 598h) [Reset = 00000000h]

PCS22MSTID is shown in [Table 3-202](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register22

**Table 3-202. PCS22MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS45MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS44MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.197 PCS23MSTID Register (Offset = 59Ch) [Reset = 00000000h]

PCS23MSTID is shown in [Table 3-203](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register23

**Table 3-203. PCS23MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS47MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS46MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.198 PCS24MSTID Register (Offset = 5A0h) [Reset = 00000000h]

PCS24MSTID is shown in [Table 3-204](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register24

**Table 3-204. PCS24MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS49MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS48MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.199 PCS25MSTID Register (Offset = 5A4h) [Reset = 00000000h]

PCS25MSTID is shown in [Table 3-205](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register25

**Table 3-205. PCS25MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS51MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS50MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.200 PCS26MSTID Register (Offset = 5A8h) [Reset = 00000000h]

PCS26MSTID is shown in [Table 3-206](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register26

**Table 3-206. PCS26MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS53MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS52MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.201 PCS27MSTID Register (Offset = 5ACh) [Reset = 00000000h]

PCS27MSTID is shown in [Table 3-207](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register27

**Table 3-207. PCS27MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS55MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS54MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.202 PCS28MSTID Register (Offset = 5B0h) [Reset = 00000000h]

PCS28MSTID is shown in [Table 3-208](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register28

**Table 3-208. PCS28MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS57MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS56MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.1.203 PCS29MSTID Register (Offset = 5B4h) [Reset = 00000000h]

PCS29MSTID is shown in [Table 3-209](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register29

**Table 3-209. PCS29MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS59MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS58MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.204 PCS30MSTID Register (Offset = 5B8h) [Reset = 00000000h]

PCS30MSTID is shown in [Table 3-210](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register30

**Table 3-210. PCS30MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS61MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS60MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.205 PCS31MSTID Register (Offset = 5BCh) [Reset = 00000000h]

PCS31MSTID is shown in [Table 3-211](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register31

**Table 3-211. PCS31MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS63MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS62MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.1.206 PPCS0MSTID Register (Offset = 5C0h) [Reset = 00000000h]

PPCS0MSTID is shown in [Table 3-212](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register32

**Table 3-212. PPCS0MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS1MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS0MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.207 PPCS1MSTID Register (Offset = 5C4h) [Reset = 00000000h]

PPCS1MSTID is shown in [Table 3-213](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register33

**Table 3-213. PPCS1MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS3MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS2MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.208 PPCS2MSTID Register (Offset = 5C8h) [Reset = 00000000h]

PPCS2MSTID is shown in [Table 3-214](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register<sup>34</sup>

**Table 3-214. PPCS2MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS5MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS4MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.209 PPCS3MSTID Register (Offset = 5CCh) [Reset = 00000000h]

PPCS3MSTID is shown in [Table 3-215](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register35

**Table 3-215. PPCS3MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS7MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS6MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.210 PPCS4MSTID Register (Offset = 5D0h) [Reset = 00000000h]

PPCS4MSTID is shown in [Table 3-216](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register36

**Table 3-216. PPCS4MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS9MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS8MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.1.211 PPCS5MSTID Register (Offset = 5D4h) [Reset = 00000000h]

PPCS5MSTID is shown in [Table 3-217](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register37

**Table 3-217. PPCS5MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS11MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS10MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.212 PPCS6MSTID Register (Offset = 5D8h) [Reset = 00000000h]

PPCS6MSTID is shown in [Table 3-218](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register38

**Table 3-218. PPCS6MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS13MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS12MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.213 PPCS7MSTID Register (Offset = 5DCh) [Reset = 00000000h]

PPCS7MSTID is shown in [Table 3-219](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register39

**Table 3-219. PPCS7MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS15MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS14MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.1.214 PCREXTMSTID Register (Offset = 5E0h) [Reset = 00000000h]

PCREXTMSTID is shown in [Table 3-220](#).

Return to the [Summary Table](#).

Master-ID Protection Register for external PCR

**Table 3-220. PCREXTMSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PCREXT_MSTID	R/W	0h	These bits sets the permission for maximum of 16 masters to address the external PCR frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2 MSS\_PCR2 Registers

Table 3-221 lists the memory-mapped registers for the MSS\_PCR2 registers. All register offset addresses not listed in Table 3-221 should be considered as reserved locations and the register contents should not be modified.

**Table 3-221. MSS\_PCR2 Registers**

Offset	Acronym	Register Name	Section
0h	PMPROTSET0	PMPROTSET0	<a href="#">Section 3.4.2.1</a>
4h	PMPROTSET1	PMPROTSET1	<a href="#">Section 3.4.2.2</a>
10h	PMPROTCLR0	PMPROTCLR0	<a href="#">Section 3.4.2.3</a>
14h	PMPROTCLR1	PMPROTCLR1	<a href="#">Section 3.4.2.4</a>
20h	PPROTSET_0	PPROTSET_0	<a href="#">Section 3.4.2.5</a>
24h	PPROTSET_1	PPROTSET_1	<a href="#">Section 3.4.2.6</a>
28h	PPROTSET_2	PPROTSET_2	<a href="#">Section 3.4.2.7</a>
2Ch	PPROTSET_3	PPROTSET_3	<a href="#">Section 3.4.2.8</a>
40h	PPROTCLR0	PPROTCLR0	<a href="#">Section 3.4.2.9</a>
44h	PPROTCLR1	PPROTCLR1	<a href="#">Section 3.4.2.10</a>
48h	PPROTCLR2	PPROTCLR2	<a href="#">Section 3.4.2.11</a>
4Ch	PPROTCLR3	PPROTCLR3	<a href="#">Section 3.4.2.12</a>
60h	PCSPWRDWNSET0	PCSPWRDWNSET0	<a href="#">Section 3.4.2.13</a>
64h	PCSPWRDWNSET1	PCSPWRDWNSET1	<a href="#">Section 3.4.2.14</a>
70h	PCSPWRDWNCLR0	PCSPWRDWNCLR0	<a href="#">Section 3.4.2.15</a>
74h	PCSPWRDWNCLR1	PCSPWRDWNCLR1	<a href="#">Section 3.4.2.16</a>
80h	PSPWRDWNSET0	PSPWRDWNSET0	<a href="#">Section 3.4.2.17</a>
84h	PSPWRDWNSET1	PSPWRDWNSET1	<a href="#">Section 3.4.2.18</a>
88h	PSPWRDWNSET2	PSPWRDWNSET2	<a href="#">Section 3.4.2.19</a>
8Ch	PSPWRDWNSET3	PSPWRDWNSET3	<a href="#">Section 3.4.2.20</a>
A0h	PSPWRDWNCLR0	PSPWRDWNCLR0	<a href="#">Section 3.4.2.21</a>
A4h	PSPWRDWNCLR1	PSPWRDWNCLR1	<a href="#">Section 3.4.2.22</a>
A8h	PSPWRDWNCLR2	PSPWRDWNCLR2	<a href="#">Section 3.4.2.23</a>
ACh	PSPWRDWNCLR3	PSPWRDWNCLR3	<a href="#">Section 3.4.2.24</a>
C0h	PDPWRDWNSET	PDPWRDWNSET	<a href="#">Section 3.4.2.25</a>
C4h	PDPWRDWNCLR	PDPWRDWNCLR	<a href="#">Section 3.4.2.26</a>
200h	MSTIDWRENA	MSTIDWRENA	<a href="#">Section 3.4.2.27</a>
204h	MSTIDENA	MSTIDENA	<a href="#">Section 3.4.2.28</a>
208h	MSTIDDIAGCTRL	MSTIDDIAGCTRL	<a href="#">Section 3.4.2.29</a>
300h	PS0MSTID_L	PS0MSTID_L	<a href="#">Section 3.4.2.30</a>
304h	PS0MSTID_H	PS0MSTID_H	<a href="#">Section 3.4.2.31</a>
308h	PS1MSTID_L	PS1MSTID_L	<a href="#">Section 3.4.2.32</a>
30Ch	PS1MSTID_H	PS1MSTID_H	<a href="#">Section 3.4.2.33</a>
310h	PS2MSTID_L	PS2MSTID_L	<a href="#">Section 3.4.2.34</a>
314h	PS2MSTID_H	PS2MSTID_H	<a href="#">Section 3.4.2.35</a>
318h	PS3MSTID_L	PS3MSTID_L	<a href="#">Section 3.4.2.36</a>
31Ch	PS3MSTID_H	PS3MSTID_H	<a href="#">Section 3.4.2.37</a>
320h	PS4MSTID_L	PS4MSTID_L	<a href="#">Section 3.4.2.38</a>
324h	PS4MSTID_H	PS4MSTID_H	<a href="#">Section 3.4.2.39</a>
328h	PS5MSTID_L	PS5MSTID_L	<a href="#">Section 3.4.2.40</a>

**Table 3-221. MSS\_PCR2 Registers (continued)**

Offset	Acronym	Register Name	Section
32Ch	PS5MSTID_H	PS5MSTID_H	<a href="#">Section 3.4.2.41</a>
330h	PS6MSTID_L	PS6MSTID_L	<a href="#">Section 3.4.2.42</a>
334h	PS6MSTID_H	PS6MSTID_H	<a href="#">Section 3.4.2.43</a>
338h	PS7MSTID_L	PS7MSTID_L	<a href="#">Section 3.4.2.44</a>
33Ch	PS7MSTID_H	PS7MSTID_H	<a href="#">Section 3.4.2.45</a>
340h	PS8MSTID_L	PS8MSTID_L	<a href="#">Section 3.4.2.46</a>
344h	PS8MSTID_H	PS8MSTID_H	<a href="#">Section 3.4.2.47</a>
348h	PS9MSTID_L	PS9MSTID_L	<a href="#">Section 3.4.2.48</a>
34Ch	PS9MSTID_H	PS9MSTID_H	<a href="#">Section 3.4.2.49</a>
350h	PS10MSTID_L	PS10MSTID_L	<a href="#">Section 3.4.2.50</a>
354h	PS10MSTID_H	PS10MSTID_H	<a href="#">Section 3.4.2.51</a>
358h	PS11MSTID_L	PS11MSTID_L	<a href="#">Section 3.4.2.52</a>
35Ch	PS11MSTID_H	PS11MSTID_H	<a href="#">Section 3.4.2.53</a>
360h	PS12MSTID_L	PS12MSTID_L	<a href="#">Section 3.4.2.54</a>
364h	PS12MSTID_H	PS12MSTID_H	<a href="#">Section 3.4.2.55</a>
368h	PS13MSTID_L	PS13MSTID_L	<a href="#">Section 3.4.2.56</a>
36Ch	PS13MSTID_H	PS13MSTID_H	<a href="#">Section 3.4.2.57</a>
370h	PS14MSTID_L	PS14MSTID_L	<a href="#">Section 3.4.2.58</a>
374h	PS14MSTID_H	PS14MSTID_H	<a href="#">Section 3.4.2.59</a>
378h	PS15MSTID_L	PS15MSTID_L	<a href="#">Section 3.4.2.60</a>
37Ch	PS15MSTID_H	PS15MSTID_H	<a href="#">Section 3.4.2.61</a>
380h	PS16MSTID_L	PS16MSTID_L	<a href="#">Section 3.4.2.62</a>
384h	PS16MSTID_H	PS16MSTID_H	<a href="#">Section 3.4.2.63</a>
388h	PS17MSTID_L	PS17MSTID_L	<a href="#">Section 3.4.2.64</a>
38Ch	PS17MSTID_H	PS17MSTID_H	<a href="#">Section 3.4.2.65</a>
390h	PS18MSTID_L	PS18MSTID_L	<a href="#">Section 3.4.2.66</a>
394h	PS18MSTID_H	PS18MSTID_H	<a href="#">Section 3.4.2.67</a>
398h	PS19MSTID_L	PS19MSTID_L	<a href="#">Section 3.4.2.68</a>
39Ch	PS19MSTID_H	PS19MSTID_H	<a href="#">Section 3.4.2.69</a>
3A0h	PS20MSTID_L	PS20MSTID_L	<a href="#">Section 3.4.2.70</a>
3A4h	PS20MSTID_H	PS20MSTID_H	<a href="#">Section 3.4.2.71</a>
3A8h	PS21MSTID_L	PS21MSTID_L	<a href="#">Section 3.4.2.72</a>
3ACh	PS21MSTID_H	PS21MSTID_H	<a href="#">Section 3.4.2.73</a>
3B0h	PS22MSTID_L	PS22MSTID_L	<a href="#">Section 3.4.2.74</a>
3B4h	PS22MSTID_H	PS22MSTID_H	<a href="#">Section 3.4.2.75</a>
3B8h	PS23MSTID_L	PS23MSTID_L	<a href="#">Section 3.4.2.76</a>
3BCh	PS23MSTID_H	PS23MSTID_H	<a href="#">Section 3.4.2.77</a>
3C0h	PS24MSTID_L	PS24MSTID_L	<a href="#">Section 3.4.2.78</a>
3C4h	PS24MSTID_H	PS24MSTID_H	<a href="#">Section 3.4.2.79</a>
3C8h	PS25MSTID_L	PS25MSTID_L	<a href="#">Section 3.4.2.80</a>
3CCh	PS25MSTID_H	PS25MSTID_H	<a href="#">Section 3.4.2.81</a>
3D0h	PS26MSTID_L	PS26MSTID_L	<a href="#">Section 3.4.2.82</a>
3D4h	PS26MSTID_H	PS26MSTID_H	<a href="#">Section 3.4.2.83</a>
3D8h	PS27MSTID_L	PS27MSTID_L	<a href="#">Section 3.4.2.84</a>
3DCh	PS27MSTID_H	PS27MSTID_H	<a href="#">Section 3.4.2.85</a>

**Table 3-221. MSS\_PCR2 Registers (continued)**

Offset	Acronym	Register Name	Section
3E0h	PS28MSTID_L	PS28MSTID_L	<a href="#">Section 3.4.2.86</a>
3E4h	PS28MSTID_H	PS28MSTID_H	<a href="#">Section 3.4.2.87</a>
3E8h	PS29MSTID_L	PS29MSTID_L	<a href="#">Section 3.4.2.88</a>
3ECh	PS29MSTID_H	PS29MSTID_H	<a href="#">Section 3.4.2.89</a>
3F0h	PS30MSTID_L	PS30MSTID_L	<a href="#">Section 3.4.2.90</a>
3F4h	PS30MSTID_H	PS30MSTID_H	<a href="#">Section 3.4.2.91</a>
3F8h	PS31MSTID_L	PS31MSTID_L	<a href="#">Section 3.4.2.92</a>
3FCh	PS31MSTID_H	PS31MSTID_H	<a href="#">Section 3.4.2.93</a>
400h	PPS0MSTID_L	PPS0MSTID_L	<a href="#">Section 3.4.2.94</a>
404h	PPS0MSTID_H	PPS0MSTID_H	<a href="#">Section 3.4.2.95</a>
408h	PPS1MSTID_L	PPS1MSTID_L	<a href="#">Section 3.4.2.96</a>
40Ch	PPS1MSTID_H	PPS1MSTID_H	<a href="#">Section 3.4.2.97</a>
410h	PPS2MSTID_L	PPS2MSTID_L	<a href="#">Section 3.4.2.98</a>
414h	PPS2MSTID_H	PPS2MSTID_H	<a href="#">Section 3.4.2.99</a>
418h	PPS3MSTID_L	PPS3MSTID_L	<a href="#">Section 3.4.2.100</a>
41Ch	PPS3MSTID_H	PPS3MSTID_H	<a href="#">Section 3.4.2.101</a>
420h	PPS4MSTID_L	PPS4MSTID_L	<a href="#">Section 3.4.2.102</a>
424h	PPS4MSTID_H	PPS4MSTID_H	<a href="#">Section 3.4.2.103</a>
428h	PPS5MSTID_L	PPS5MSTID_L	<a href="#">Section 3.4.2.104</a>
42Ch	PPS5MSTID_H	PPS5MSTID_H	<a href="#">Section 3.4.2.105</a>
430h	PPS6MSTID_L	PPS6MSTID_L	<a href="#">Section 3.4.2.106</a>
434h	PPS6MSTID_H	PPS6MSTID_H	<a href="#">Section 3.4.2.107</a>
438h	PPS7MSTID_L	PPS7MSTID_L	<a href="#">Section 3.4.2.108</a>
43Ch	PPS7MSTID_H	PPS7MSTID_H	<a href="#">Section 3.4.2.109</a>
440h	PPSE0MSTID_L	PPSE0MSTID_L	<a href="#">Section 3.4.2.110</a>
444h	PPSE0MSTID_H	PPSE0MSTID_H	<a href="#">Section 3.4.2.111</a>
448h	PPSE1MSTID_L	PPSE1MSTID_L	<a href="#">Section 3.4.2.112</a>
44Ch	PPSE1MSTID_H	PPSE1MSTID_H	<a href="#">Section 3.4.2.113</a>
450h	PPSE2MSTID_L	PPSE2MSTID_L	<a href="#">Section 3.4.2.114</a>
454h	PPSE2MSTID_H	PPSE2MSTID_H	<a href="#">Section 3.4.2.115</a>
458h	PPSE3MSTID_L	PPSE3MSTID_L	<a href="#">Section 3.4.2.116</a>
45Ch	PPSE3MSTID_H	PPSE3MSTID_H	<a href="#">Section 3.4.2.117</a>
460h	PPSE4MSTID_L	PPSE4MSTID_L	<a href="#">Section 3.4.2.118</a>
464h	PPSE4MSTID_H	PPSE4MSTID_H	<a href="#">Section 3.4.2.119</a>
468h	PPSE5MSTID_L	PPSE5MSTID_L	<a href="#">Section 3.4.2.120</a>
46Ch	PPSE5MSTID_H	PPSE5MSTID_H	<a href="#">Section 3.4.2.121</a>
470h	PPSE6MSTID_L	PPSE6MSTID_L	<a href="#">Section 3.4.2.122</a>
474h	PPSE6MSTID_H	PPSE6MSTID_H	<a href="#">Section 3.4.2.123</a>
478h	PPSE7MSTID_L	PPSE7MSTID_L	<a href="#">Section 3.4.2.124</a>
47Ch	PPSE7MSTID_H	PPSE7MSTID_H	<a href="#">Section 3.4.2.125</a>
480h	PPSE8MSTID_L	PPSE8MSTID_L	<a href="#">Section 3.4.2.126</a>
484h	PPSE8MSTID_H	PPSE8MSTID_H	<a href="#">Section 3.4.2.127</a>
488h	PPSE9MSTID_L	PPSE9MSTID_L	<a href="#">Section 3.4.2.128</a>
48Ch	PPSE9MSTID_H	PPSE9MSTID_H	<a href="#">Section 3.4.2.129</a>
490h	PPSE10MSTID_L	PPSE10MSTID_L	<a href="#">Section 3.4.2.130</a>

**Table 3-221. MSS\_PCR2 Registers (continued)**

Offset	Acronym	Register Name	Section
494h	PPSE10MSTID_H	PPSE10MSTID_H	<a href="#">Section 3.4.2.131</a>
498h	PPSE11MSTID_L	PPSE11MSTID_L	<a href="#">Section 3.4.2.132</a>
49Ch	PPSE11MSTID_H	PPSE11MSTID_H	<a href="#">Section 3.4.2.133</a>
4A0h	PPSE12MSTID_L	PPSE12MSTID_L	<a href="#">Section 3.4.2.134</a>
4A4h	PPSE12MSTID_H	PPSE12MSTID_H	<a href="#">Section 3.4.2.135</a>
4A8h	PPSE13MSTID_L	PPSE13MSTID_L	<a href="#">Section 3.4.2.136</a>
4ACh	PPSE13MSTID_H	PPSE13MSTID_H	<a href="#">Section 3.4.2.137</a>
4B0h	PPSE14MSTID_L	PPSE14MSTID_L	<a href="#">Section 3.4.2.138</a>
4B4h	PPSE14MSTID_H	PPSE14MSTID_H	<a href="#">Section 3.4.2.139</a>
4B8h	PPSE15MSTID_L	PPSE15MSTID_L	<a href="#">Section 3.4.2.140</a>
4BCh	PPSE15MSTID_H	PPSE15MSTID_H	<a href="#">Section 3.4.2.141</a>
4C0h	PPSE16MSTID_L	PPSE16MSTID_L	<a href="#">Section 3.4.2.142</a>
4C4h	PPSE16MSTID_H	PPSE16MSTID_H	<a href="#">Section 3.4.2.143</a>
4C8h	PPSE17MSTID_L	PPSE17MSTID_L	<a href="#">Section 3.4.2.144</a>
4CCh	PPSE17MSTID_H	PPSE17MSTID_H	<a href="#">Section 3.4.2.145</a>
4D0h	PPSE18MSTID_L	PPSE18MSTID_L	<a href="#">Section 3.4.2.146</a>
4D4h	PPSE18MSTID_H	PPSE18MSTID_H	<a href="#">Section 3.4.2.147</a>
4D8h	PPSE19MSTID_L	PPSE19MSTID_L	<a href="#">Section 3.4.2.148</a>
4DCh	PPSE19MSTID_H	PPSE19MSTID_H	<a href="#">Section 3.4.2.149</a>
4E0h	PPSE20MSTID_L	PPSE20MSTID_L	<a href="#">Section 3.4.2.150</a>
4E4h	PPSE20MSTID_H	PPSE20MSTID_H	<a href="#">Section 3.4.2.151</a>
4E8h	PPSE21MSTID_L	PPSE21MSTID_L	<a href="#">Section 3.4.2.152</a>
4ECh	PPSE21MSTID_H	PPSE21MSTID_H	<a href="#">Section 3.4.2.153</a>
4F0h	PPSE22MSTID_L	PPSE22MSTID_L	<a href="#">Section 3.4.2.154</a>
4F4h	PPSE22MSTID_H	PPSE22MSTID_H	<a href="#">Section 3.4.2.155</a>
4F8h	PPSE23MSTID_L	PPSE23MSTID_L	<a href="#">Section 3.4.2.156</a>
4FCh	PPSE23MSTID_H	PPSE23MSTID_H	<a href="#">Section 3.4.2.157</a>
500h	PPSE24MSTID_L	PPSE24MSTID_L	<a href="#">Section 3.4.2.158</a>
504h	PPSE24MSTID_H	PPSE24MSTID_H	<a href="#">Section 3.4.2.159</a>
508h	PPSE25MSTID_L	PPSE25MSTID_L	<a href="#">Section 3.4.2.160</a>
50Ch	PPSE25MSTID_H	PPSE25MSTID_H	<a href="#">Section 3.4.2.161</a>
510h	PPSE26MSTID_L	PPSE26MSTID_L	<a href="#">Section 3.4.2.162</a>
514h	PPSE26MSTID_H	PPSE26MSTID_H	<a href="#">Section 3.4.2.163</a>
518h	PPSE27MSTID_L	PPSE27MSTID_L	<a href="#">Section 3.4.2.164</a>
51Ch	PPSE27MSTID_H	PPSE27MSTID_H	<a href="#">Section 3.4.2.165</a>
520h	PPSE28MSTID_L	PPSE28MSTID_L	<a href="#">Section 3.4.2.166</a>
524h	PPSE28MSTID_H	PPSE28MSTID_H	<a href="#">Section 3.4.2.167</a>
528h	PPSE29MSTID_L	PPSE29MSTID_L	<a href="#">Section 3.4.2.168</a>
52Ch	PPSE29MSTID_H	PPSE29MSTID_H	<a href="#">Section 3.4.2.169</a>
530h	PPSE30MSTID_L	PPSE30MSTID_L	<a href="#">Section 3.4.2.170</a>
534h	PPSE30MSTID_H	PPSE30MSTID_H	<a href="#">Section 3.4.2.171</a>
538h	PPSE31MSTID_L	PPSE31MSTID_L	<a href="#">Section 3.4.2.172</a>
53Ch	PPSE31MSTID_H	PPSE31MSTID_H	<a href="#">Section 3.4.2.173</a>
540h	PCS0MSTID	PCS0MSTID	<a href="#">Section 3.4.2.174</a>
544h	PCS1MSTID	PCS1MSTID	<a href="#">Section 3.4.2.175</a>



**Table 3-221. MSS\_PCR2 Registers (continued)**

Offset	Acronym	Register Name	Section
548h	PCS2MSTID	PCS2MSTID	<a href="#">Section 3.4.2.176</a>
54Ch	PCS3MSTID	PCS3MSTID	<a href="#">Section 3.4.2.177</a>
550h	PCS4MSTID	PCS4MSTID	<a href="#">Section 3.4.2.178</a>
554h	PCS5MSTID	PCS5MSTID	<a href="#">Section 3.4.2.179</a>
558h	PCS6MSTID	PCS6MSTID	<a href="#">Section 3.4.2.180</a>
55Ch	PCS7MSTID	PCS7MSTID	<a href="#">Section 3.4.2.181</a>
560h	PCS8MSTID	PCS8MSTID	<a href="#">Section 3.4.2.182</a>
564h	PCS9MSTID	PCS9MSTID	<a href="#">Section 3.4.2.183</a>
568h	PCS10MSTID	PCS10MSTID	<a href="#">Section 3.4.2.184</a>
56Ch	PCS11MSTID	PCS11MSTID	<a href="#">Section 3.4.2.185</a>
570h	PCS12MSTID	PCS12MSTID	<a href="#">Section 3.4.2.186</a>
574h	PCS13MSTID	PCS13MSTID	<a href="#">Section 3.4.2.187</a>
578h	PCS14MSTID	PCS14MSTID	<a href="#">Section 3.4.2.188</a>
57Ch	PCS15MSTID	PCS15MSTID	<a href="#">Section 3.4.2.189</a>
580h	PCS16MSTID	PCS16MSTID	<a href="#">Section 3.4.2.190</a>
584h	PCS17MSTID	PCS17MSTID	<a href="#">Section 3.4.2.191</a>
588h	PCS18MSTID	PCS18MSTID	<a href="#">Section 3.4.2.192</a>
58Ch	PCS19MSTID	PCS19MSTID	<a href="#">Section 3.4.2.193</a>
590h	PCS20MSTID	PCS20MSTID	<a href="#">Section 3.4.2.194</a>
594h	PCS21MSTID	PCS21MSTID	<a href="#">Section 3.4.2.195</a>
598h	PCS22MSTID	PCS22MSTID	<a href="#">Section 3.4.2.196</a>
59Ch	PCS23MSTID	PCS23MSTID	<a href="#">Section 3.4.2.197</a>
5A0h	PCS24MSTID	PCS24MSTID	<a href="#">Section 3.4.2.198</a>
5A4h	PCS25MSTID	PCS25MSTID	<a href="#">Section 3.4.2.199</a>
5A8h	PCS26MSTID	PCS26MSTID	<a href="#">Section 3.4.2.200</a>
5ACh	PCS27MSTID	PCS27MSTID	<a href="#">Section 3.4.2.201</a>
5B0h	PCS28MSTID	PCS28MSTID	<a href="#">Section 3.4.2.202</a>
5B4h	PCS29MSTID	PCS29MSTID	<a href="#">Section 3.4.2.203</a>
5B8h	PCS30MSTID	PCS30MSTID	<a href="#">Section 3.4.2.204</a>
5BCh	PCS31MSTID	PCS31MSTID	<a href="#">Section 3.4.2.205</a>
5C0h	PPCS0MSTID	PPCS0MSTID	<a href="#">Section 3.4.2.206</a>
5C4h	PPCS1MSTID	PPCS1MSTID	<a href="#">Section 3.4.2.207</a>
5C8h	PPCS2MSTID	PPCS2MSTID	<a href="#">Section 3.4.2.208</a>
5CCh	PPCS3MSTID	PPCS3MSTID	<a href="#">Section 3.4.2.209</a>
5D0h	PPCS4MSTID	PPCS4MSTID	<a href="#">Section 3.4.2.210</a>
5D4h	PPCS5MSTID	PPCS5MSTID	<a href="#">Section 3.4.2.211</a>
5D8h	PPCS6MSTID	PPCS6MSTID	<a href="#">Section 3.4.2.212</a>
5DCh	PPCS7MSTID	PPCS7MSTID	<a href="#">Section 3.4.2.213</a>
5E0h	PCREXTMSTID	PCREXTMSTID	<a href="#">Section 3.4.2.214</a>

Complex bit access types are encoded to fit into small table cells. [Table 3-222](#) shows the codes that are used for access types in this section.

**Table 3-222. MSS\_PCR2 Access Type Codes**

Access Type	Code	Description
Read Type		

**Table 3-222. MSS\_PCR2 Access Type Codes  
(continued)**

Access Type	Code	Description
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 3.4.2.1 PMPROTSET0 Register (Offset = 0h) [Reset = 00000000h]

PMPROTSET0 is shown in [Table 3-223](#).

Return to the [Summary Table](#).

Set-only register to protect PCS frames 0 to 31

**Table 3-223. PMPROTSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
30	PCS30_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
29	PCS29_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
28	PCS28_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
27	PCS27_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-223. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
26	PCS26_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
25	PCS25_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
24	PCS24_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
23	PCS23_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
22	PCS22_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
21	PCS21_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-223. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
20	PCS20_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
19	PCS19_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
18	PCS18_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
17	PCS17_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
16	PCS16_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
15	PCS15_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-223. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	PCS14_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
13	PCS13_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
12	PCS12_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
11	PCS11_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
10	PCS10_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
9	PCS9_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-223. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS8_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
7	PCS7_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
6	PCS6_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
5	PCS5_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
4	PCS4_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
3	PCS3_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.



**Table 3-223. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
1	PCS1_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
0	PCS0_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.2 PMPROTSET1 Register (Offset = 4h) [Reset = 00000000h]

PMPROTSET1 is shown in [Table 3-224](#).

Return to the [Summary Table](#).

Set-only register to protect PCS frames 32 to 63

**Table 3-224. PMPROTSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
30	PCS62_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
29	PCS61_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
28	PCS60_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
27	PCS59_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
26	PCS58_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
25	PCS57_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-224. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS56_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
23	PCS55_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
22	PCS54_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
21	PCS53_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
20	PCS52_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
19	PCS51_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
18	PCS50_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
17	PCS49_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-224. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS48_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
15	PCS47_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
14	PCS46_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
13	PCS45_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
12	PCS44_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
11	PCS43_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
10	PCS42_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
9	PCS41_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-224. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS40_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
7	PCS39_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
6	PCS38_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
5	PCS37_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
4	PCS36_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
3	PCS35_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
2	PCS34_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
1	PCS33_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-224. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS32_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

### 3.4.2.3 PMPROTCLR0 Register (Offset = 10h) [Reset = 00000000h]

PMPROTCLR0 is shown in [Table 3-225](#).

Return to the [Summary Table](#).

Clear-only register to protect PCS frames 0 to 31

**Table 3-225. PMPROTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
30	PCS30_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
29	PCS29_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
28	PCS28_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
27	PCS27_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
26	PCS26_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
25	PCS25_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

**Table 3-225. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS24_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
23	PCS23_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
22	PCS22_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
21	PCS21_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
20	PCS20_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
19	PCS19_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
18	PCS18_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
17	PCS17_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect



**Table 3-225. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS16_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
15	PCS15_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
14	PCS14_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
13	PCS13_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
12	PCS12_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
11	PCS11_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
10	PCS10_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
9	PCS9_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect



**Table 3-225. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS8_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
7	PCS7_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
6	PCS6_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
5	PCS5_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
4	PCS4_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
3	PCS3_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
2	PCS2_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
1	PCS1_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

**Table 3-225. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS0_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

### 3.4.2.4 PMPROTCLR1 Register (Offset = 14h) [Reset = 00000000h]

PMPROTCLR1 is shown in [Table 3-226](#).

Return to the [Summary Table](#).

Clear-only register to protect PCS frames 32 to 63

**Table 3-226. PMPROTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
30	PCS62_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
29	PCS61_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
28	PCS60_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
27	PCS59_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
26	PCS58_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
25	PCS57_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-226. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS56_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
23	PCS55_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
22	PCS54_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
21	PCS53_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
20	PCS52_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
19	PCS51_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
18	PCS50_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
17	PCS49_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-226. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS48_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
15	PCS47_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
14	PCS46_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
13	PCS45_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
12	PCS44_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
11	PCS43_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
10	PCS42_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
9	PCS41_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-226. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS40_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
7	PCS39_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
6	PCS38_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
5	PCS37_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
4	PCS36_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
3	PCS35_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
2	PCS34_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
1	PCS33_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-226. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS32_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect



### 3.4.2.5 PPROTSET\_0 Register (Offset = 20h) [Reset = 00000000h]

PPROTSET\_0 is shown in [Table 3-227](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS0 to PS7

**Table 3-227. PPROTSET\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



**Table 3-227. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-227. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-227. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-227. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

### 3.4.2.6 PPROTSET\_1 Register (Offset = 24h) [Reset = 00000000h]

PPROTSET\_1 is shown in [Table 3-228](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS8 to PS15

**Table 3-228. PPROTSET\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-228. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-228. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect



**Table 3-228. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect



**Table 3-228. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

### 3.4.2.7 PPROTSET\_2 Register (Offset = 28h) [Reset = 00000000h]

PPROTSET\_2 is shown in [Table 3-229](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS16 to PS23

**Table 3-229. PPROTSET\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-229. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-229. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-229. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-229. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

### 3.4.2.8 PPROTSET\_3 Register (Offset = 2Ch) [Reset = 00000000h]

PPROTSET\_3 is shown in [Table 3-230](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS24 to PS31

**Table 3-230. PPROTSET\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



**Table 3-230. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



**Table 3-230. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-230. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-230. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

### 3.4.2.9 PPROTCLR0 Register (Offset = 40h) [Reset = 00000000h]

PPROTCLR0 is shown in [Table 3-231](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS0 to PS7

**Table 3-231. PPROTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-231. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-231. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-231. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



**Table 3-231. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



### 3.4.2.10 PPROTCLR1 Register (Offset = 44h) [Reset = 00000000h]

PPROTCLR1 is shown in [Table 3-232](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS8 to PS15

**Table 3-232. PPROTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-232. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-232. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-232. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-232. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

### 3.4.2.11 PPROTCLR2 Register (Offset = 48h) [Reset = 00000000h]

PPROTCLR2 is shown in [Table 3-233](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS16 to PS23

**Table 3-233. PPROTCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-233. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect



**Table 3-233. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect



**Table 3-233. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-233. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

### 3.4.2.12 PPROTCLR3 Register (Offset = 4Ch) [Reset = 00000000h]

PPROTCLR3 is shown in [Table 3-234](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS24 to PS31

**Table 3-234. PPROTCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-234. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-234. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-234. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-234. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



### 3.4.2.13 PCSPWRDWNSET0 Register (Offset = 60h) [Reset = 00000000h]

PCSPWRDWNSET0 is shown in [Table 3-235](#).

Return to the [Summary Table](#).

Set-only register to powerdown independent (non-shared) PCS frames 0 to 31

**Table 3-235. PCSPWRDWNSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-235. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS22_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-235. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS12_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-235. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.2.14 PCSPWRDWNSET1 Register (Offset = 64h) [Reset = 00000000h]

PCSPWRDWNSET1 is shown in [Table 3-236](#).

Return to the [Summary Table](#).

Set-only register to powerdown independent (non-shared) PCS frames 32 to 63

**Table 3-236. PCSPWRDWNSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-236. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS54_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-236. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS44_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-236. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS34_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect



### 3.4.2.15 PCSPWRDWNCLR0 Register (Offset = 70h) [Reset = 00000000h]

PCSPWRDWNCLR0 is shown in [Table 3-237](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 0 to 31

**Table 3-237. PCSPWRDWNCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-237. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS22_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-237. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS12_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-237. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.2.16 PCSPWRDWNCLR1 Register (Offset = 74h) [Reset = 00000000h]

PCSPWRDWNCLR1 is shown in [Table 3-238](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 32 to 63

**Table 3-238. PCSPWRDWNCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-238. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS54_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-238. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS44_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-238. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS34_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect



### 3.4.2.17 PSPWRDWNSET0 Register (Offset = 80h) [Reset = 00000000h]

PSPWRDWNSET0 is shown in [Table 3-239](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS0 to PS7

**Table 3-239. PSPWRDWNSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-239. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-239. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable bit only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-239. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-239. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.2.18 PSPWRDWNSET1 Register (Offset = 84h) [Reset = 00000000h]

PSPWRDWNSET1 is shown in [Table 3-240](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS8 to PS15

**Table 3-240. PSPWRDWNSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-240. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-240. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect



**Table 3-240. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect



**Table 3-240. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

### 3.4.2.19 PSPWRDWNSET2 Register (Offset = 88h) [Reset = 00000000h]

PSPWRDWNSET2 is shown in [Table 3-241](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS16 to PS23

**Table 3-241. PSPWRDWNSET2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-241. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-241. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-241. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-241. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

### 3.4.2.20 PSPWRDWNSET3 Register (Offset = 8Ch) [Reset = 00000000h]

PSPWRDWNSET3 is shown in [Table 3-242](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS24 to PS31

**Table 3-242. PSPWRDWNSET3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-242. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect



**Table 3-242. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-242. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-242. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

### 3.4.2.21 PSPWRDWNCLR0 Register (Offset = A0h) [Reset = 00000000h]

PSPWRDWNCLR0 is shown in [Table 3-243](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS0 to PS7

**Table 3-243. PSPWRDWNCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-243. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-243. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-243. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-243. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect



### 3.4.2.22 PSPWRDWNCLR1 Register (Offset = A4h) [Reset = 00000000h]

PSPWRDWNCLR1 is shown in [Table 3-244](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS8 to PS15

**Table 3-244. PSPWRDWNCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-244. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-244. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-244. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-244. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

### 3.4.2.23 PSPWRDWNCLR2 Register (Offset = A8h) [Reset = 00000000h]

PSPWRDWNCLR2 is shown in [Table 3-245](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS16 to PS23

**Table 3-245. PSPWRDWNCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-245. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect



**Table 3-245. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect



**Table 3-245. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-245. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

### 3.4.2.24 PSPWRDWNCLR3 Register (Offset = ACh) [Reset = 00000000h]

PSPWRDWNCLR3 is shown in [Table 3-246](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS24 to PS31

**Table 3-246. PSPWRDWNCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_CLR	R/W	0h	
30	PS31_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-246. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
23	PS29_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-246. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
15	PS27_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-246. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7	PS25_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

### 3.4.2.25 PDPWRDWNSET Register (Offset = C0h) [Reset = 00000000h]

PDPWRDWNSET is shown in [Table 3-247](#).

Return to the [Summary Table](#).

Set-only register to powerdown the debug frame

**Table 3-247. PDPWRDWNSET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = Clock to the debug frame needs to be powered down. 0 = Clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get set in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect

### 3.4.2.26 PDPWRDWNCLR Register (Offset = C4h) [Reset = 00000000h]

PDPWRDWNCLR is shown in [Table 3-248](#).

Return to the [Summary Table](#).

Clear-only register to deassert the debug frame's powerdown bit

**Table 3-248. PDPWRDWNCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the debug frame needs to be powered down. 0 = The clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get cleared in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect



### 3.4.2.27 MSTIDWRENA Register (Offset = 200h) [Reset = 00000000h]

MSTIDWRENA is shown in [Table 3-249](#).

Return to the [Summary Table](#).

MasterID Protection Write Enable Register

**Table 3-249. MSTIDWRENA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTIDREG_WRENA	R/W	0h	Readable in both user and privileged modes. 1010 = All master-id registers are unlocked and available for write. others = Writes to all master-id registers are locked. Writable only in privileged mode 1010 = Writes to master-id registers are unlocked. others = Writes to master-id registers are locked.

### 3.4.2.28 MSTIDENA Register (Offset = 204h) [Reset = 00000000h]

MSTIDENA is shown in [Table 3-250](#).

Return to the [Summary Table](#).

MasterID Protection Enable Register

**Table 3-250. MSTIDENA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTID_CHK_EN	R/W	0h	Readable in both user and privileged modes. Writable only in privileged mode 1010 = Enable the master-id feature check. others = Master-id check is disabled.

### 3.4.2.29 MSTIDDIAGCTRL Register (Offset = 208h) [Reset = 00000000h]

MSTIDDIAGCTRL is shown in [Table 3-251](#).

Return to the [Summary Table](#).

MasterID Diagnostic Control Register

**Table 3-251. MSTIDDIAGCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-8	DIAG_CMP_VALUE	R/W	0h	MasterID diagnostic mode control register bits; 4-bit data which is compared with the master-id register of all defined frames during diagnostic mode. Any error in compare logic is indicated through AERROR output from PCR. Readable in both user and privileged modes. Reads the programmed value in diagnostic compare value field. Writable only in privileged mode
7-4	RESERVED	R	0h	Reserved
3-0	DIAG_MODE_EN	R/W	0h	MasterID compare logic diagnostic mode enable bits; 4-bit key for enabling the master-id registers compare logic. Readable in both user and privileged modes. Writable only in privileged mode 1010 = Master-id compare diagnostic mode is enabled. others = Master-id compare diagnostic mode is disabled.

### 3.4.2.30 PS0MSTID\_L Register (Offset = 300h) [Reset = 00000000h]

PS0MSTID\_L is shown in [Table 3-252](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register0\_L

**Table 3-252. PS0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.31 PS0MSTID\_H Register (Offset = 304h) [Reset = 00000000h]

PS0MSTID\_H is shown in [Table 3-253](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register0\_H

**Table 3-253. PS0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.32 PS1MSTID\_L Register (Offset = 308h) [Reset = 00000000h]

PS1MSTID\_L is shown in [Table 3-254](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register1\_L

**Table 3-254. PS1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.33 PS1MSTID\_H Register (Offset = 30Ch) [Reset = 00000000h]

PS1MSTID\_H is shown in [Table 3-255](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register1\_H

**Table 3-255. PS1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.34 PS2MSTID\_L Register (Offset = 310h) [Reset = 00000000h]

PS2MSTID\_L is shown in [Table 3-256](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register2\_L

**Table 3-256. PS2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.35 PS2MSTID\_H Register (Offset = 314h) [Reset = 00000000h]

PS2MSTID\_H is shown in [Table 3-257](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register2\_H

**Table 3-257. PS2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.36 PS3MSTID\_L Register (Offset = 318h) [Reset = 00000000h]

PS3MSTID\_L is shown in [Table 3-258](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register3\_L

**Table 3-258. PS3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.37 PS3MSTID\_H Register (Offset = 31Ch) [Reset = 00000000h]

PS3MSTID\_H is shown in [Table 3-259](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register3\_H

**Table 3-259. PS3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.38 PS4MSTID\_L Register (Offset = 320h) [Reset = 00000000h]

PS4MSTID\_L is shown in [Table 3-260](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register4\_L

**Table 3-260. PS4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.39 PS4MSTID\_H Register (Offset = 324h) [Reset = 00000000h]

PS4MSTID\_H is shown in [Table 3-261](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register4\_H

**Table 3-261. PS4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.40 PS5MSTID\_L Register (Offset = 328h) [Reset = 00000000h]

PS5MSTID\_L is shown in [Table 3-262](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register5\_L

**Table 3-262. PS5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.41 PS5MSTID\_H Register (Offset = 32Ch) [Reset = 00000000h]

PS5MSTID\_H is shown in [Table 3-263](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register5\_H

**Table 3-263. PS5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.42 PS6MSTID\_L Register (Offset = 330h) [Reset = 00000000h]

PS6MSTID\_L is shown in [Table 3-264](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register6\_L

**Table 3-264. PS6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.43 PS6MSTID\_H Register (Offset = 334h) [Reset = 00000000h]

PS6MSTID\_H is shown in [Table 3-265](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register6\_H

**Table 3-265. PS6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.44 PS7MSTID\_L Register (Offset = 338h) [Reset = 00000000h]

PS7MSTID\_L is shown in [Table 3-266](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register7\_L

**Table 3-266. PS7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.45 PS7MSTID\_H Register (Offset = 33Ch) [Reset = 00000000h]

PS7MSTID\_H is shown in [Table 3-267](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register7\_H

**Table 3-267. PS7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.46 PS8MSTID\_L Register (Offset = 340h) [Reset = 00000000h]

PS8MSTID\_L is shown in [Table 3-268](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register8\_L

**Table 3-268. PS8MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.47 PS8MSTID\_H Register (Offset = 344h) [Reset = 00000000h]

PS8MSTID\_H is shown in [Table 3-269](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register8\_H

**Table 3-269. PS8MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.48 PS9MSTID\_L Register (Offset = 348h) [Reset = 00000000h]

PS9MSTID\_L is shown in [Table 3-270](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register9\_L

**Table 3-270. PS9MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.49 PS9MSTID\_H Register (Offset = 34Ch) [Reset = 00000000h]

PS9MSTID\_H is shown in [Table 3-271](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register9\_H

**Table 3-271. PS9MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.50 PS10MSTID\_L Register (Offset = 350h) [Reset = 00000000h]

PS10MSTID\_L is shown in [Table 3-272](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register10\_L

**Table 3-272. PS10MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.51 PS10MSTID\_H Register (Offset = 354h) [Reset = 00000000h]

PS10MSTID\_H is shown in [Table 3-273](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register10\_H

**Table 3-273. PS10MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.52 PS11MSTID\_L Register (Offset = 358h) [Reset = 00000000h]

PS11MSTID\_L is shown in [Table 3-274](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register11\_L

**Table 3-274. PS11MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.53 PS11MSTID\_H Register (Offset = 35Ch) [Reset = 00000000h]

PS11MSTID\_H is shown in [Table 3-275](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register11\_H

**Table 3-275. PS11MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.54 PS12MSTID\_L Register (Offset = 360h) [Reset = 00000000h]

PS12MSTID\_L is shown in [Table 3-276](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register12\_L

**Table 3-276. PS12MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.55 PS12MSTID\_H Register (Offset = 364h) [Reset = 00000000h]

PS12MSTID\_H is shown in [Table 3-277](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register12\_H

**Table 3-277. PS12MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.56 PS13MSTID\_L Register (Offset = 368h) [Reset = 00000000h]

PS13MSTID\_L is shown in [Table 3-278](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register13\_L

**Table 3-278. PS13MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.57 PS13MSTID\_H Register (Offset = 36Ch) [Reset = 00000000h]

PS13MSTID\_H is shown in [Table 3-279](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register13\_H

**Table 3-279. PS13MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.58 PS14MSTID\_L Register (Offset = 370h) [Reset = 00000000h]

PS14MSTID\_L is shown in [Table 3-280](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register14\_L

**Table 3-280. PS14MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.59 PS14MSTID\_H Register (Offset = 374h) [Reset = 00000000h]

PS14MSTID\_H is shown in [Table 3-281](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register14\_H

**Table 3-281. PS14MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.60 PS15MSTID\_L Register (Offset = 378h) [Reset = 00000000h]

PS15MSTID\_L is shown in [Table 3-282](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register15\_L

**Table 3-282. PS15MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.61 PS15MSTID\_H Register (Offset = 37Ch) [Reset = 00000000h]

PS15MSTID\_H is shown in [Table 3-283](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register15\_H

**Table 3-283. PS15MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.62 PS16MSTID\_L Register (Offset = 380h) [Reset = 00000000h]

PS16MSTID\_L is shown in [Table 3-284](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register16\_L

**Table 3-284. PS16MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.63 PS16MSTID\_H Register (Offset = 384h) [Reset = 00000000h]

PS16MSTID\_H is shown in [Table 3-285](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register16\_H

**Table 3-285. PS16MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.64 PS17MSTID\_L Register (Offset = 388h) [Reset = 00000000h]

PS17MSTID\_L is shown in [Table 3-286](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register17\_L

**Table 3-286. PS17MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.65 PS17MSTID\_H Register (Offset = 38Ch) [Reset = 00000000h]

PS17MSTID\_H is shown in [Table 3-287](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register17\_H

**Table 3-287. PS17MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.66 PS18MSTID\_L Register (Offset = 390h) [Reset = 00000000h]

PS18MSTID\_L is shown in [Table 3-288](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register18\_L

**Table 3-288. PS18MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.67 PS18MSTID\_H Register (Offset = 394h) [Reset = 00000000h]

PS18MSTID\_H is shown in [Table 3-289](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register18\_H

**Table 3-289. PS18MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.68 PS19MSTID\_L Register (Offset = 398h) [Reset = 00000000h]

PS19MSTID\_L is shown in [Table 3-290](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register19\_L

**Table 3-290. PS19MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.69 PS19MSTID\_H Register (Offset = 39Ch) [Reset = 00000000h]

PS19MSTID\_H is shown in [Table 3-291](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register19\_H

**Table 3-291. PS19MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.70 PS20MSTID\_L Register (Offset = 3A0h) [Reset = 00000000h]

PS20MSTID\_L is shown in [Table 3-292](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register20\_L

**Table 3-292. PS20MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.71 PS20MSTID\_H Register (Offset = 3A4h) [Reset = 00000000h]

PS20MSTID\_H is shown in [Table 3-293](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register20\_H

**Table 3-293. PS20MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.72 PS21MSTID\_L Register (Offset = 3A8h) [Reset = 00000000h]

PS21MSTID\_L is shown in [Table 3-294](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register21\_L

**Table 3-294. PS21MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.73 PS21MSTID\_H Register (Offset = 3ACh) [Reset = 00000000h]

PS21MSTID\_H is shown in [Table 3-295](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register21\_H

**Table 3-295. PS21MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.74 PS22MSTID\_L Register (Offset = 3B0h) [Reset = 00000000h]

PS22MSTID\_L is shown in [Table 3-296](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register22\_L

**Table 3-296. PS22MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.75 PS22MSTID\_H Register (Offset = 3B4h) [Reset = 00000000h]

PS22MSTID\_H is shown in [Table 3-297](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register22\_H

**Table 3-297. PS22MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.76 PS23MSTID\_L Register (Offset = 3B8h) [Reset = 00000000h]

PS23MSTID\_L is shown in [Table 3-298](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register23\_L

**Table 3-298. PS23MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.77 PS23MSTID\_H Register (Offset = 3BCh) [Reset = 00000000h]

PS23MSTID\_H is shown in [Table 3-299](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register23\_H

**Table 3-299. PS23MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.78 PS24MSTID\_L Register (Offset = 3C0h) [Reset = 00000000h]

PS24MSTID\_L is shown in [Table 3-300](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register24\_L

**Table 3-300. PS24MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.79 PS24MSTID\_H Register (Offset = 3C4h) [Reset = 00000000h]

PS24MSTID\_H is shown in [Table 3-301](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register24\_H

**Table 3-301. PS24MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.80 PS25MSTID\_L Register (Offset = 3C8h) [Reset = 00000000h]

PS25MSTID\_L is shown in [Table 3-302](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register25\_L

**Table 3-302. PS25MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.81 PS25MSTID\_H Register (Offset = 3CCh) [Reset = 00000000h]

PS25MSTID\_H is shown in [Table 3-303](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register25\_H

**Table 3-303. PS25MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.82 PS26MSTID\_L Register (Offset = 3D0h) [Reset = 00000000h]

PS26MSTID\_L is shown in [Table 3-304](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register26\_L

**Table 3-304. PS26MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.83 PS26MSTID\_H Register (Offset = 3D4h) [Reset = 00000000h]

PS26MSTID\_H is shown in [Table 3-305](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register26\_H

**Table 3-305. PS26MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.84 PS27MSTID\_L Register (Offset = 3D8h) [Reset = 00000000h]

PS27MSTID\_L is shown in [Table 3-306](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register27\_L

**Table 3-306. PS27MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.85 PS27MSTID\_H Register (Offset = 3DCh) [Reset = 00000000h]

PS27MSTID\_H is shown in [Table 3-307](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register27\_H

**Table 3-307. PS27MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.86 PS28MSTID\_L Register (Offset = 3E0h) [Reset = 00000000h]

PS28MSTID\_L is shown in [Table 3-308](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register28\_L

**Table 3-308. PS28MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.87 PS28MSTID\_H Register (Offset = 3E4h) [Reset = 00000000h]

PS28MSTID\_H is shown in [Table 3-309](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register28\_H

**Table 3-309. PS28MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.88 PS29MSTID\_L Register (Offset = 3E8h) [Reset = 00000000h]

PS29MSTID\_L is shown in [Table 3-310](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register29\_L

**Table 3-310. PS29MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.89 PS29MSTID\_H Register (Offset = 3ECh) [Reset = 00000000h]

PS29MSTID\_H is shown in [Table 3-311](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register29\_H

**Table 3-311. PS29MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.90 PS30MSTID\_L Register (Offset = 3F0h) [Reset = 00000000h]

PS30MSTID\_L is shown in [Table 3-312](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register30\_L

**Table 3-312. PS30MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.91 PS30MSTID\_H Register (Offset = 3F4h) [Reset = 00000000h]

PS30MSTID\_H is shown in [Table 3-313](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register30\_H

**Table 3-313. PS30MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.92 PS31MSTID\_L Register (Offset = 3F8h) [Reset = 00000000h]

PS31MSTID\_L is shown in [Table 3-314](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register31\_L

**Table 3-314. PS31MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.93 PS31MSTID\_H Register (Offset = 3FCh) [Reset = 00000000h]

PS31MSTID\_H is shown in [Table 3-315](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register31\_H

**Table 3-315. PS31MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.94 PPS0MSTID\_L Register (Offset = 400h) [Reset = 00000000h]

PPS0MSTID\_L is shown in [Table 3-316](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register0\_L

**Table 3-316. PPS0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.95 PPS0MSTID\_H Register (Offset = 404h) [Reset = 00000000h]

PPS0MSTID\_H is shown in [Table 3-317](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register0\_H

**Table 3-317. PPS0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.96 PPS1MSTID\_L Register (Offset = 408h) [Reset = 00000000h]

PPS1MSTID\_L is shown in [Table 3-318](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register1\_L

**Table 3-318. PPS1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.97 PPS1MSTID\_H Register (Offset = 40Ch) [Reset = 00000000h]

PPS1MSTID\_H is shown in [Table 3-319](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register1\_H

**Table 3-319. PPS1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.98 PPS2MSTID\_L Register (Offset = 410h) [Reset = 00000000h]

PPS2MSTID\_L is shown in [Table 3-320](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register2\_L

**Table 3-320. PPS2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.99 PPS2MSTID\_H Register (Offset = 414h) [Reset = 00000000h]

PPS2MSTID\_H is shown in [Table 3-321](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register2\_H

**Table 3-321. PPS2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.100 PPS3MSTID\_L Register (Offset = 418h) [Reset = 00000000h]

PPS3MSTID\_L is shown in [Table 3-322](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register3\_L

**Table 3-322. PPS3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.101 PPS3MSTID\_H Register (Offset = 41Ch) [Reset = 00000000h]

PPS3MSTID\_H is shown in [Table 3-323](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register3\_H

**Table 3-323. PPS3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.102 PPS4MSTID\_L Register (Offset = 420h) [Reset = 00000000h]

PPS4MSTID\_L is shown in [Table 3-324](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register4\_L

**Table 3-324. PPS4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.103 PPS4MSTID\_H Register (Offset = 424h) [Reset = 00000000h]

PPS4MSTID\_H is shown in [Table 3-325](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register4\_H

**Table 3-325. PPS4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.104 PPS5MSTID\_L Register (Offset = 428h) [Reset = 00000000h]

PPS5MSTID\_L is shown in [Table 3-326](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register5\_L

**Table 3-326. PPS5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.105 PPS5MSTID\_H Register (Offset = 42Ch) [Reset = 00000000h]

PPS5MSTID\_H is shown in [Table 3-327](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register5\_H

**Table 3-327. PPS5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.106 PPS6MSTID\_L Register (Offset = 430h) [Reset = 00000000h]

PPS6MSTID\_L is shown in [Table 3-328](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register6\_L

**Table 3-328. PPS6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.107 PPS6MSTID\_H Register (Offset = 434h) [Reset = 00000000h]

PPS6MSTID\_H is shown in [Table 3-329](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register6\_H

**Table 3-329. PPS6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.108 PPS7MSTID\_L Register (Offset = 438h) [Reset = 00000000h]

PPS7MSTID\_L is shown in [Table 3-330](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register7\_L

**Table 3-330. PPS7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.109 PPS7MSTID\_H Register (Offset = 43Ch) [Reset = 00000000h]

PPS7MSTID\_H is shown in [Table 3-331](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register7\_H

**Table 3-331. PPS7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.110 PPSE0MSTID\_L Register (Offset = 440h) [Reset = 00000000h]

PPSE0MSTID\_L is shown in [Table 3-332](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register0\_L

**Table 3-332. PPSE0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.111 PPSE0MSTID\_H Register (Offset = 444h) [Reset = 00000000h]

PPSE0MSTID\_H is shown in [Table 3-333](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register0\_H

**Table 3-333. PPSE0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.112 PPSE1MSTID\_L Register (Offset = 448h) [Reset = 00000000h]

PPSE1MSTID\_L is shown in [Table 3-334](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register1\_L

**Table 3-334. PPSE1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.113 PPSE1MSTID\_H Register (Offset = 44Ch) [Reset = 00000000h]

PPSE1MSTID\_H is shown in [Table 3-335](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register1\_H

**Table 3-335. PPSE1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.114 PPSE2MSTID\_L Register (Offset = 450h) [Reset = 00000000h]

PPSE2MSTID\_L is shown in [Table 3-336](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register2\_L

**Table 3-336. PPSE2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.115 PPSE2MSTID\_H Register (Offset = 454h) [Reset = 00000000h]

PPSE2MSTID\_H is shown in [Table 3-337](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register2\_H

**Table 3-337. PPSE2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.116 PPSE3MSTID\_L Register (Offset = 458h) [Reset = 00000000h]

PPSE3MSTID\_L is shown in [Table 3-338](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register3\_L

**Table 3-338. PPSE3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.117 PPSE3MSTID\_H Register (Offset = 45Ch) [Reset = 00000000h]

PPSE3MSTID\_H is shown in [Table 3-339](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register3\_H

**Table 3-339. PPSE3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.118 PPSE4MSTID\_L Register (Offset = 460h) [Reset = 00000000h]

PPSE4MSTID\_L is shown in [Table 3-340](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register4\_L

**Table 3-340. PPSE4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.119 PPSE4MSTID\_H Register (Offset = 464h) [Reset = 00000000h]

PPSE4MSTID\_H is shown in [Table 3-341](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register4\_H

**Table 3-341. PPSE4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.120 PPSE5MSTID\_L Register (Offset = 468h) [Reset = 00000000h]

PPSE5MSTID\_L is shown in [Table 3-342](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register5\_L

**Table 3-342. PPSE5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.121 PPSE5MSTID\_H Register (Offset = 46Ch) [Reset = 00000000h]

PPSE5MSTID\_H is shown in [Table 3-343](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register5\_H

**Table 3-343. PPSE5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.122 PPSE6MSTID\_L Register (Offset = 470h) [Reset = 00000000h]

PPSE6MSTID\_L is shown in [Table 3-344](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register6\_L

**Table 3-344. PPSE6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.123 PPSE6MSTID\_H Register (Offset = 474h) [Reset = 00000000h]

PPSE6MSTID\_H is shown in [Table 3-345](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register6\_H

**Table 3-345. PPSE6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.124 PPSE7MSTID\_L Register (Offset = 478h) [Reset = 00000000h]

PPSE7MSTID\_L is shown in [Table 3-346](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register7\_L

**Table 3-346. PPSE7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.125 PPSE7MSTID\_H Register (Offset = 47Ch) [Reset = 00000000h]

PPSE7MSTID\_H is shown in [Table 3-347](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register7\_H

**Table 3-347. PPSE7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.126 PPSE8MSTID\_L Register (Offset = 480h) [Reset = 00000000h]

PPSE8MSTID\_L is shown in [Table 3-348](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register8\_L

**Table 3-348. PPSE8MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.127 PPSE8MSTID\_H Register (Offset = 484h) [Reset = 00000000h]

PPSE8MSTID\_H is shown in [Table 3-349](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register8\_H

**Table 3-349. PPSE8MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.128 PPSE9MSTID\_L Register (Offset = 488h) [Reset = 00000000h]

PPSE9MSTID\_L is shown in [Table 3-350](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register9\_L

**Table 3-350. PPSE9MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.129 PPSE9MSTID\_H Register (Offset = 48Ch) [Reset = 00000000h]

PPSE9MSTID\_H is shown in [Table 3-351](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register9\_H

**Table 3-351. PPSE9MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.130 PPSE10MSTID\_L Register (Offset = 490h) [Reset = 00000000h]

PPSE10MSTID\_L is shown in [Table 3-352](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register10\_L

**Table 3-352. PPSE10MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.131 PPSE10MSTID\_H Register (Offset = 494h) [Reset = 00000000h]

PPSE10MSTID\_H is shown in [Table 3-353](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register10\_H

**Table 3-353. PPSE10MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.132 PPSE11MSTID\_L Register (Offset = 498h) [Reset = 00000000h]

PPSE11MSTID\_L is shown in [Table 3-354](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register11\_L

**Table 3-354. PPSE11MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.133 PPSE11MSTID\_H Register (Offset = 49Ch) [Reset = 00000000h]

PPSE11MSTID\_H is shown in [Table 3-355](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register11\_H

**Table 3-355. PPSE11MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.134 PPSE12MSTID\_L Register (Offset = 4A0h) [Reset = 00000000h]

PPSE12MSTID\_L is shown in [Table 3-356](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register12\_L

**Table 3-356. PPSE12MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.135 PPSE12MSTID\_H Register (Offset = 4A4h) [Reset = 00000000h]

PPSE12MSTID\_H is shown in [Table 3-357](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register12\_H

**Table 3-357. PPSE12MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.136 PPSE13MSTID\_L Register (Offset = 4A8h) [Reset = 00000000h]

PPSE13MSTID\_L is shown in [Table 3-358](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register13\_L

**Table 3-358. PPSE13MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.137 PPSE13MSTID\_H Register (Offset = 4ACh) [Reset = 00000000h]

PPSE13MSTID\_H is shown in [Table 3-359](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register13\_H

**Table 3-359. PPSE13MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.138 PPSE14MSTID\_L Register (Offset = 4B0h) [Reset = 00000000h]

PPSE14MSTID\_L is shown in [Table 3-360](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register14\_L

**Table 3-360. PPSE14MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.139 PPSE14MSTID\_H Register (Offset = 4B4h) [Reset = 00000000h]

PPSE14MSTID\_H is shown in [Table 3-361](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register14\_H

**Table 3-361. PPSE14MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.140 PPSE15MSTID\_L Register (Offset = 4B8h) [Reset = 00000000h]

PPSE15MSTID\_L is shown in [Table 3-362](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register15\_L

**Table 3-362. PPSE15MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.141 PPSE15MSTID\_H Register (Offset = 4BCh) [Reset = 00000000h]

PPSE15MSTID\_H is shown in [Table 3-363](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register15\_H

**Table 3-363. PPSE15MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.142 PPSE16MSTID\_L Register (Offset = 4C0h) [Reset = 00000000h]

PPSE16MSTID\_L is shown in [Table 3-364](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register16\_L

**Table 3-364. PPSE16MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.143 PPSE16MSTID\_H Register (Offset = 4C4h) [Reset = 00000000h]

PPSE16MSTID\_H is shown in [Table 3-365](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register16\_H

**Table 3-365. PPSE16MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.144 PPSE17MSTID\_L Register (Offset = 4C8h) [Reset = 00000000h]

PPSE17MSTID\_L is shown in [Table 3-366](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register17\_L

**Table 3-366. PPSE17MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.145 PPSE17MSTID\_H Register (Offset = 4CCh) [Reset = 00000000h]

PPSE17MSTID\_H is shown in [Table 3-367](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register17\_H

**Table 3-367. PPSE17MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.146 PPSE18MSTID\_L Register (Offset = 4D0h) [Reset = 00000000h]

PPSE18MSTID\_L is shown in [Table 3-368](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register18\_L

**Table 3-368. PPSE18MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.147 PPSE18MSTID\_H Register (Offset = 4D4h) [Reset = 00000000h]

PPSE18MSTID\_H is shown in [Table 3-369](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register18\_H

**Table 3-369. PPSE18MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.148 PPSE19MSTID\_L Register (Offset = 4D8h) [Reset = 00000000h]

PPSE19MSTID\_L is shown in [Table 3-370](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register19\_L

**Table 3-370. PPSE19MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.149 PPSE19MSTID\_H Register (Offset = 4DCh) [Reset = 00000000h]

PPSE19MSTID\_H is shown in [Table 3-371](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register19\_H

**Table 3-371. PPSE19MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.150 PPSE20MSTID\_L Register (Offset = 4E0h) [Reset = 00000000h]

PPSE20MSTID\_L is shown in [Table 3-372](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register20\_L

**Table 3-372. PPSE20MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.151 PPSE20MSTID\_H Register (Offset = 4E4h) [Reset = 00000000h]

PPSE20MSTID\_H is shown in [Table 3-373](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register20\_H

**Table 3-373. PPSE20MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.152 PPSE21MSTID\_L Register (Offset = 4E8h) [Reset = 00000000h]

PPSE21MSTID\_L is shown in [Table 3-374](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register21\_L

**Table 3-374. PPSE21MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.153 PPSE21MSTID\_H Register (Offset = 4ECh) [Reset = 00000000h]

PPSE21MSTID\_H is shown in [Table 3-375](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register21\_H

**Table 3-375. PPSE21MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.154 PPSE22MSTID\_L Register (Offset = 4F0h) [Reset = 00000000h]

PPSE22MSTID\_L is shown in [Table 3-376](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register22\_L

**Table 3-376. PPSE22MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.155 PPSE22MSTID\_H Register (Offset = 4F4h) [Reset = 00000000h]

PPSE22MSTID\_H is shown in [Table 3-377](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register22\_H

**Table 3-377. PPSE22MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.156 PPSE23MSTID\_L Register (Offset = 4F8h) [Reset = 00000000h]

PPSE23MSTID\_L is shown in [Table 3-378](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register23\_L

**Table 3-378. PPSE23MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.157 PPSE23MSTID\_H Register (Offset = 4FCh) [Reset = 00000000h]

PPSE23MSTID\_H is shown in [Table 3-379](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register23\_H

**Table 3-379. PPSE23MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.158 PPSE24MSTID\_L Register (Offset = 500h) [Reset = 00000000h]

PPSE24MSTID\_L is shown in [Table 3-380](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register24\_L

**Table 3-380. PPSE24MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.159 PPSE24MSTID\_H Register (Offset = 504h) [Reset = 00000000h]

PPSE24MSTID\_H is shown in [Table 3-381](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register24\_H

**Table 3-381. PPSE24MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.160 PPSE25MSTID\_L Register (Offset = 508h) [Reset = 00000000h]

PPSE25MSTID\_L is shown in [Table 3-382](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register25\_L

**Table 3-382. PPSE25MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.161 PPSE25MSTID\_H Register (Offset = 50Ch) [Reset = 00000000h]

PPSE25MSTID\_H is shown in [Table 3-383](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register25\_H

**Table 3-383. PPSE25MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.162 PPSE26MSTID\_L Register (Offset = 510h) [Reset = 00000000h]

PPSE26MSTID\_L is shown in [Table 3-384](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register26\_L

**Table 3-384. PPSE26MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.163 PPSE26MSTID\_H Register (Offset = 514h) [Reset = 00000000h]

PPSE26MSTID\_H is shown in [Table 3-385](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register26\_H

**Table 3-385. PPSE26MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.164 PPSE27MSTID\_L Register (Offset = 518h) [Reset = 00000000h]

PPSE27MSTID\_L is shown in [Table 3-386](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register27\_L

**Table 3-386. PPSE27MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.165 PPSE27MSTID\_H Register (Offset = 51Ch) [Reset = 00000000h]

PPSE27MSTID\_H is shown in [Table 3-387](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register27\_H

**Table 3-387. PPSE27MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.166 PPSE28MSTID\_L Register (Offset = 520h) [Reset = 00000000h]

PPSE28MSTID\_L is shown in [Table 3-388](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register28\_L

**Table 3-388. PPSE28MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.167 PPSE28MSTID\_H Register (Offset = 524h) [Reset = 00000000h]

PPSE28MSTID\_H is shown in [Table 3-389](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register28\_H

**Table 3-389. PPSE28MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.168 PPSE29MSTID\_L Register (Offset = 528h) [Reset = 00000000h]

PPSE29MSTID\_L is shown in [Table 3-390](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register29\_L

**Table 3-390. PPSE29MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.169 PPSE29MSTID\_H Register (Offset = 52Ch) [Reset = 00000000h]

PPSE29MSTID\_H is shown in [Table 3-391](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register29\_H

**Table 3-391. PPSE29MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.170 PPSE30MSTID\_L Register (Offset = 530h) [Reset = 00000000h]

PPSE30MSTID\_L is shown in [Table 3-392](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register30\_L

**Table 3-392. PPSE30MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE30_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.171 PPSE30MSTID\_H Register (Offset = 534h) [Reset = 00000000h]

PPSE30MSTID\_H is shown in [Table 3-393](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register30\_H

**Table 3-393. PPSE30MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.172 PPSE31MSTID\_L Register (Offset = 538h) [Reset = 00000000h]

PPSE31MSTID\_L is shown in [Table 3-394](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register31\_L

**Table 3-394. PPSE31MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.173 PPSE31MSTID\_H Register (Offset = 53Ch) [Reset = 00000000h]

PPSE31MSTID\_H is shown in [Table 3-395](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register31\_H

**Table 3-395. PPSE31MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.174 PCS0MSTID Register (Offset = 540h) [Reset = 00000000h]

PCS0MSTID is shown in [Table 3-396](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register0

**Table 3-396. PCS0MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS1MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS0MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.175 PCS1MSTID Register (Offset = 544h) [Reset = 00000000h]

PCS1MSTID is shown in [Table 3-397](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register1

**Table 3-397. PCS1MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS3MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS2MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.176 PCS2MSTID Register (Offset = 548h) [Reset = 00000000h]

PCS2MSTID is shown in [Table 3-398](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register2

**Table 3-398. PCS2MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS5MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS4MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.177 PCS3MSTID Register (Offset = 54Ch) [Reset = 00000000h]

PCS3MSTID is shown in [Table 3-399](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register3

**Table 3-399. PCS3MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS7MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS6MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.178 PCS4MSTID Register (Offset = 550h) [Reset = 00000000h]

PCS4MSTID is shown in [Table 3-400](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register4

**Table 3-400. PCS4MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS9MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS8MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.2.179 PCS5MSTID Register (Offset = 554h) [Reset = 00000000h]

PCS5MSTID is shown in [Table 3-401](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register5

**Table 3-401. PCS5MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS11MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS10MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.180 PCS6MSTID Register (Offset = 558h) [Reset = 00000000h]

PCS6MSTID is shown in [Table 3-402](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register6

**Table 3-402. PCS6MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS13MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS12MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.181 PCS7MSTID Register (Offset = 55Ch) [Reset = 00000000h]

PCS7MSTID is shown in [Table 3-403](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register7

**Table 3-403. PCS7MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS15MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS14MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.182 PCS8MSTID Register (Offset = 560h) [Reset = 00000000h]

PCS8MSTID is shown in [Table 3-404](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register8

**Table 3-404. PCS8MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS17MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS16MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.183 PCS9MSTID Register (Offset = 564h) [Reset = 00000000h]

PCS9MSTID is shown in [Table 3-405](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register9

**Table 3-405. PCS9MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS19MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS18MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.184 PCS10MSTID Register (Offset = 568h) [Reset = 00000000h]

PCS10MSTID is shown in [Table 3-406](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register10

**Table 3-406. PCS10MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS21MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS20MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.185 PCS11MSTID Register (Offset = 56Ch) [Reset = 00000000h]

PCS11MSTID is shown in [Table 3-407](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register11

**Table 3-407. PCS11MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS23MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS22MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.186 PCS12MSTID Register (Offset = 570h) [Reset = 00000000h]

PCS12MSTID is shown in [Table 3-408](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register12

**Table 3-408. PCS12MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS25MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS24MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.2.187 PCS13MSTID Register (Offset = 574h) [Reset = 00000000h]

PCS13MSTID is shown in [Table 3-409](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register13

**Table 3-409. PCS13MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS27MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS26MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.188 PCS14MSTID Register (Offset = 578h) [Reset = 00000000h]

PCS14MSTID is shown in [Table 3-410](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register14

**Table 3-410. PCS14MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS29MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS28MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.189 PCS15MSTID Register (Offset = 57Ch) [Reset = 00000000h]

PCS15MSTID is shown in [Table 3-411](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register15

**Table 3-411. PCS15MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS31MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS30MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.190 PCS16MSTID Register (Offset = 580h) [Reset = 00000000h]

PCS16MSTID is shown in [Table 3-412](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register16

**Table 3-412. PCS16MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS33MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS32MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.191 PCS17MSTID Register (Offset = 584h) [Reset = 00000000h]

PCS17MSTID is shown in [Table 3-413](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register17

**Table 3-413. PCS17MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS35MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS34MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.192 PCS18MSTID Register (Offset = 588h) [Reset = 00000000h]

PCS18MSTID is shown in [Table 3-414](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register18

**Table 3-414. PCS18MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS37MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS36MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.193 PCS19MSTID Register (Offset = 58Ch) [Reset = 00000000h]

PCS19MSTID is shown in [Table 3-415](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register19

**Table 3-415. PCS19MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS39MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS38MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.194 PCS20MSTID Register (Offset = 590h) [Reset = 00000000h]

PCS20MSTID is shown in [Table 3-416](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register20

**Table 3-416. PCS20MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS41MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS40MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.2.195 PCS21MSTID Register (Offset = 594h) [Reset = 00000000h]

PCS21MSTID is shown in [Table 3-417](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register21

**Table 3-417. PCS21MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS43MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS42MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.196 PCS22MSTID Register (Offset = 598h) [Reset = 00000000h]

PCS22MSTID is shown in [Table 3-418](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register22

**Table 3-418. PCS22MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS45MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS44MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.197 PCS23MSTID Register (Offset = 59Ch) [Reset = 00000000h]

PCS23MSTID is shown in [Table 3-419](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register23

**Table 3-419. PCS23MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS47MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS46MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.198 PCS24MSTID Register (Offset = 5A0h) [Reset = 00000000h]

PCS24MSTID is shown in [Table 3-420](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register24

**Table 3-420. PCS24MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS49MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS48MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.199 PCS25MSTID Register (Offset = 5A4h) [Reset = 00000000h]

PCS25MSTID is shown in [Table 3-421](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register25

**Table 3-421. PCS25MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS51MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS50MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.200 PCS26MSTID Register (Offset = 5A8h) [Reset = 00000000h]

PCS26MSTID is shown in [Table 3-422](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register26

**Table 3-422. PCS26MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS53MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS52MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.201 PCS27MSTID Register (Offset = 5ACh) [Reset = 00000000h]

PCS27MSTID is shown in [Table 3-423](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register27

**Table 3-423. PCS27MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS55MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS54MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.202 PCS28MSTID Register (Offset = 5B0h) [Reset = 00000000h]

PCS28MSTID is shown in [Table 3-424](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register28

**Table 3-424. PCS28MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS57MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS56MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.2.203 PCS29MSTID Register (Offset = 5B4h) [Reset = 00000000h]

PCS29MSTID is shown in [Table 3-425](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register29

**Table 3-425. PCS29MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS59MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS58MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.204 PCS30MSTID Register (Offset = 5B8h) [Reset = 00000000h]

PCS30MSTID is shown in [Table 3-426](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register30

**Table 3-426. PCS30MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS61MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS60MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.205 PCS31MSTID Register (Offset = 5BCh) [Reset = 00000000h]

PCS31MSTID is shown in [Table 3-427](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register31

**Table 3-427. PCS31MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS63MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS62MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.2.206 PPCS0MSTID Register (Offset = 5C0h) [Reset = 00000000h]

PPCS0MSTID is shown in [Table 3-428](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register32

**Table 3-428. PPCS0MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS1MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS0MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.207 PPCS1MSTID Register (Offset = 5C4h) [Reset = 00000000h]

PPCS1MSTID is shown in [Table 3-429](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register33

**Table 3-429. PPCS1MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS3MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS2MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.208 PPCS2MSTID Register (Offset = 5C8h) [Reset = 00000000h]

PPCS2MSTID is shown in [Table 3-430](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register<sup>34</sup>

**Table 3-430. PPCS2MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS5MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS4MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.209 PPCS3MSTID Register (Offset = 5CCh) [Reset = 00000000h]

PPCS3MSTID is shown in [Table 3-431](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register35

**Table 3-431. PPCS3MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS7MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS6MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.210 PPCS4MSTID Register (Offset = 5D0h) [Reset = 00000000h]

PPCS4MSTID is shown in [Table 3-432](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register36

**Table 3-432. PPCS4MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS9MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS8MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.2.211 PPCS5MSTID Register (Offset = 5D4h) [Reset = 00000000h]

PPCS5MSTID is shown in [Table 3-433](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register37

**Table 3-433. PPCS5MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS11MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS10MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.212 PPCS6MSTID Register (Offset = 5D8h) [Reset = 00000000h]

PPCS6MSTID is shown in [Table 3-434](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register38

**Table 3-434. PPCS6MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS13MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS12MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.213 PPCS7MSTID Register (Offset = 5DCh) [Reset = 00000000h]

PPCS7MSTID is shown in [Table 3-435](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register39

**Table 3-435. PPCS7MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS15MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS14MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.2.214 PCREXTMSTID Register (Offset = 5E0h) [Reset = 00000000h]

PCREXTMSTID is shown in [Table 3-436](#).

Return to the [Summary Table](#).

Master-ID Protection Register for external PCR

**Table 3-436. PCREXTMSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PCREXT_MSTID	R/W	0h	These bits sets the permission for maximum of 16 masters to address the external PCR frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3 DSS\_PCR Registers

Table 3-437 lists the memory-mapped registers for the DSS\_PCR registers. All register offset addresses not listed in Table 3-437 should be considered as reserved locations and the register contents should not be modified.

**Table 3-437. DSS\_PCR Registers**

Offset	Acronym	Register Name	Section
0h	PMPROTSET0	PMPROTSET0	<a href="#">Section 3.4.3.1</a>
4h	PMPROTSET1	PMPROTSET1	<a href="#">Section 3.4.3.2</a>
10h	PMPROTCLR0	PMPROTCLR0	<a href="#">Section 3.4.3.3</a>
14h	PMPROTCLR1	PMPROTCLR1	<a href="#">Section 3.4.3.4</a>
20h	PPROTSET_0	PPROTSET_0	<a href="#">Section 3.4.3.5</a>
24h	PPROTSET_1	PPROTSET_1	<a href="#">Section 3.4.3.6</a>
28h	PPROTSET_2	PPROTSET_2	<a href="#">Section 3.4.3.7</a>
2Ch	PPROTSET_3	PPROTSET_3	<a href="#">Section 3.4.3.8</a>
40h	PPROTCLR0	PPROTCLR0	<a href="#">Section 3.4.3.9</a>
44h	PPROTCLR1	PPROTCLR1	<a href="#">Section 3.4.3.10</a>
48h	PPROTCLR2	PPROTCLR2	<a href="#">Section 3.4.3.11</a>
4Ch	PPROTCLR3	PPROTCLR3	<a href="#">Section 3.4.3.12</a>
60h	PCSPWRDWNSET0	PCSPWRDWNSET0	<a href="#">Section 3.4.3.13</a>
64h	PCSPWRDWNSET1	PCSPWRDWNSET1	<a href="#">Section 3.4.3.14</a>
70h	PCSPWRDWNCLR0	PCSPWRDWNCLR0	<a href="#">Section 3.4.3.15</a>
74h	PCSPWRDWNCLR1	PCSPWRDWNCLR1	<a href="#">Section 3.4.3.16</a>
80h	PSPWRDWNSET0	PSPWRDWNSET0	<a href="#">Section 3.4.3.17</a>
84h	PSPWRDWNSET1	PSPWRDWNSET1	<a href="#">Section 3.4.3.18</a>
88h	PSPWRDWNSET2	PSPWRDWNSET2	<a href="#">Section 3.4.3.19</a>
8Ch	PSPWRDWNSET3	PSPWRDWNSET3	<a href="#">Section 3.4.3.20</a>
A0h	PSPWRDWNCLR0	PSPWRDWNCLR0	<a href="#">Section 3.4.3.21</a>
A4h	PSPWRDWNCLR1	PSPWRDWNCLR1	<a href="#">Section 3.4.3.22</a>
A8h	PSPWRDWNCLR2	PSPWRDWNCLR2	<a href="#">Section 3.4.3.23</a>
ACh	PSPWRDWNCLR3	PSPWRDWNCLR3	<a href="#">Section 3.4.3.24</a>
C0h	PDPWRDWNSET	PDPWRDWNSET	<a href="#">Section 3.4.3.25</a>
C4h	PDPWRDWNCLR	PDPWRDWNCLR	<a href="#">Section 3.4.3.26</a>
200h	MSTIDWRENA	MSTIDWRENA	<a href="#">Section 3.4.3.27</a>
204h	MSTIDENA	MSTIDENA	<a href="#">Section 3.4.3.28</a>
208h	MSTIDDIAGCTRL	MSTIDDIAGCTRL	<a href="#">Section 3.4.3.29</a>
300h	PS0MSTID_L	PS0MSTID_L	<a href="#">Section 3.4.3.30</a>
304h	PS0MSTID_H	PS0MSTID_H	<a href="#">Section 3.4.3.31</a>
308h	PS1MSTID_L	PS1MSTID_L	<a href="#">Section 3.4.3.32</a>
30Ch	PS1MSTID_H	PS1MSTID_H	<a href="#">Section 3.4.3.33</a>
310h	PS2MSTID_L	PS2MSTID_L	<a href="#">Section 3.4.3.34</a>
314h	PS2MSTID_H	PS2MSTID_H	<a href="#">Section 3.4.3.35</a>
318h	PS3MSTID_L	PS3MSTID_L	<a href="#">Section 3.4.3.36</a>
31Ch	PS3MSTID_H	PS3MSTID_H	<a href="#">Section 3.4.3.37</a>
320h	PS4MSTID_L	PS4MSTID_L	<a href="#">Section 3.4.3.38</a>
324h	PS4MSTID_H	PS4MSTID_H	<a href="#">Section 3.4.3.39</a>
328h	PS5MSTID_L	PS5MSTID_L	<a href="#">Section 3.4.3.40</a>

**Table 3-437. DSS\_PCR Registers (continued)**

Offset	Acronym	Register Name	Section
32Ch	PS5MSTID_H	PS5MSTID_H	<a href="#">Section 3.4.3.41</a>
330h	PS6MSTID_L	PS6MSTID_L	<a href="#">Section 3.4.3.42</a>
334h	PS6MSTID_H	PS6MSTID_H	<a href="#">Section 3.4.3.43</a>
338h	PS7MSTID_L	PS7MSTID_L	<a href="#">Section 3.4.3.44</a>
33Ch	PS7MSTID_H	PS7MSTID_H	<a href="#">Section 3.4.3.45</a>
340h	PS8MSTID_L	PS8MSTID_L	<a href="#">Section 3.4.3.46</a>
344h	PS8MSTID_H	PS8MSTID_H	<a href="#">Section 3.4.3.47</a>
348h	PS9MSTID_L	PS9MSTID_L	<a href="#">Section 3.4.3.48</a>
34Ch	PS9MSTID_H	PS9MSTID_H	<a href="#">Section 3.4.3.49</a>
350h	PS10MSTID_L	PS10MSTID_L	<a href="#">Section 3.4.3.50</a>
354h	PS10MSTID_H	PS10MSTID_H	<a href="#">Section 3.4.3.51</a>
358h	PS11MSTID_L	PS11MSTID_L	<a href="#">Section 3.4.3.52</a>
35Ch	PS11MSTID_H	PS11MSTID_H	<a href="#">Section 3.4.3.53</a>
360h	PS12MSTID_L	PS12MSTID_L	<a href="#">Section 3.4.3.54</a>
364h	PS12MSTID_H	PS12MSTID_H	<a href="#">Section 3.4.3.55</a>
368h	PS13MSTID_L	PS13MSTID_L	<a href="#">Section 3.4.3.56</a>
36Ch	PS13MSTID_H	PS13MSTID_H	<a href="#">Section 3.4.3.57</a>
370h	PS14MSTID_L	PS14MSTID_L	<a href="#">Section 3.4.3.58</a>
374h	PS14MSTID_H	PS14MSTID_H	<a href="#">Section 3.4.3.59</a>
378h	PS15MSTID_L	PS15MSTID_L	<a href="#">Section 3.4.3.60</a>
37Ch	PS15MSTID_H	PS15MSTID_H	<a href="#">Section 3.4.3.61</a>
380h	PS16MSTID_L	PS16MSTID_L	<a href="#">Section 3.4.3.62</a>
384h	PS16MSTID_H	PS16MSTID_H	<a href="#">Section 3.4.3.63</a>
388h	PS17MSTID_L	PS17MSTID_L	<a href="#">Section 3.4.3.64</a>
38Ch	PS17MSTID_H	PS17MSTID_H	<a href="#">Section 3.4.3.65</a>
390h	PS18MSTID_L	PS18MSTID_L	<a href="#">Section 3.4.3.66</a>
394h	PS18MSTID_H	PS18MSTID_H	<a href="#">Section 3.4.3.67</a>
398h	PS19MSTID_L	PS19MSTID_L	<a href="#">Section 3.4.3.68</a>
39Ch	PS19MSTID_H	PS19MSTID_H	<a href="#">Section 3.4.3.69</a>
3A0h	PS20MSTID_L	PS20MSTID_L	<a href="#">Section 3.4.3.70</a>
3A4h	PS20MSTID_H	PS20MSTID_H	<a href="#">Section 3.4.3.71</a>
3A8h	PS21MSTID_L	PS21MSTID_L	<a href="#">Section 3.4.3.72</a>
3ACh	PS21MSTID_H	PS21MSTID_H	<a href="#">Section 3.4.3.73</a>
3B0h	PS22MSTID_L	PS22MSTID_L	<a href="#">Section 3.4.3.74</a>
3B4h	PS22MSTID_H	PS22MSTID_H	<a href="#">Section 3.4.3.75</a>
3B8h	PS23MSTID_L	PS23MSTID_L	<a href="#">Section 3.4.3.76</a>
3BCh	PS23MSTID_H	PS23MSTID_H	<a href="#">Section 3.4.3.77</a>
3C0h	PS24MSTID_L	PS24MSTID_L	<a href="#">Section 3.4.3.78</a>
3C4h	PS24MSTID_H	PS24MSTID_H	<a href="#">Section 3.4.3.79</a>
3C8h	PS25MSTID_L	PS25MSTID_L	<a href="#">Section 3.4.3.80</a>
3CCh	PS25MSTID_H	PS25MSTID_H	<a href="#">Section 3.4.3.81</a>
3D0h	PS26MSTID_L	PS26MSTID_L	<a href="#">Section 3.4.3.82</a>
3D4h	PS26MSTID_H	PS26MSTID_H	<a href="#">Section 3.4.3.83</a>
3D8h	PS27MSTID_L	PS27MSTID_L	<a href="#">Section 3.4.3.84</a>
3DCh	PS27MSTID_H	PS27MSTID_H	<a href="#">Section 3.4.3.85</a>

**Table 3-437. DSS\_PCR Registers (continued)**

Offset	Acronym	Register Name	Section
3E0h	PS28MSTID_L	PS28MSTID_L	<a href="#">Section 3.4.3.86</a>
3E4h	PS28MSTID_H	PS28MSTID_H	<a href="#">Section 3.4.3.87</a>
3E8h	PS29MSTID_L	PS29MSTID_L	<a href="#">Section 3.4.3.88</a>
3ECh	PS29MSTID_H	PS29MSTID_H	<a href="#">Section 3.4.3.89</a>
3F0h	PS30MSTID_L	PS30MSTID_L	<a href="#">Section 3.4.3.90</a>
3F4h	PS30MSTID_H	PS30MSTID_H	<a href="#">Section 3.4.3.91</a>
3F8h	PS31MSTID_L	PS31MSTID_L	<a href="#">Section 3.4.3.92</a>
3FCh	PS31MSTID_H	PS31MSTID_H	<a href="#">Section 3.4.3.93</a>
400h	PPS0MSTID_L	PPS0MSTID_L	<a href="#">Section 3.4.3.94</a>
404h	PPS0MSTID_H	PPS0MSTID_H	<a href="#">Section 3.4.3.95</a>
408h	PPS1MSTID_L	PPS1MSTID_L	<a href="#">Section 3.4.3.96</a>
40Ch	PPS1MSTID_H	PPS1MSTID_H	<a href="#">Section 3.4.3.97</a>
410h	PPS2MSTID_L	PPS2MSTID_L	<a href="#">Section 3.4.3.98</a>
414h	PPS2MSTID_H	PPS2MSTID_H	<a href="#">Section 3.4.3.99</a>
418h	PPS3MSTID_L	PPS3MSTID_L	<a href="#">Section 3.4.3.100</a>
41Ch	PPS3MSTID_H	PPS3MSTID_H	<a href="#">Section 3.4.3.101</a>
420h	PPS4MSTID_L	PPS4MSTID_L	<a href="#">Section 3.4.3.102</a>
424h	PPS4MSTID_H	PPS4MSTID_H	<a href="#">Section 3.4.3.103</a>
428h	PPS5MSTID_L	PPS5MSTID_L	<a href="#">Section 3.4.3.104</a>
42Ch	PPS5MSTID_H	PPS5MSTID_H	<a href="#">Section 3.4.3.105</a>
430h	PPS6MSTID_L	PPS6MSTID_L	<a href="#">Section 3.4.3.106</a>
434h	PPS6MSTID_H	PPS6MSTID_H	<a href="#">Section 3.4.3.107</a>
438h	PPS7MSTID_L	PPS7MSTID_L	<a href="#">Section 3.4.3.108</a>
43Ch	PPS7MSTID_H	PPS7MSTID_H	<a href="#">Section 3.4.3.109</a>
440h	PPSE0MSTID_L	PPSE0MSTID_L	<a href="#">Section 3.4.3.110</a>
444h	PPSE0MSTID_H	PPSE0MSTID_H	<a href="#">Section 3.4.3.111</a>
448h	PPSE1MSTID_L	PPSE1MSTID_L	<a href="#">Section 3.4.3.112</a>
44Ch	PPSE1MSTID_H	PPSE1MSTID_H	<a href="#">Section 3.4.3.113</a>
450h	PPSE2MSTID_L	PPSE2MSTID_L	<a href="#">Section 3.4.3.114</a>
454h	PPSE2MSTID_H	PPSE2MSTID_H	<a href="#">Section 3.4.3.115</a>
458h	PPSE3MSTID_L	PPSE3MSTID_L	<a href="#">Section 3.4.3.116</a>
45Ch	PPSE3MSTID_H	PPSE3MSTID_H	<a href="#">Section 3.4.3.117</a>
460h	PPSE4MSTID_L	PPSE4MSTID_L	<a href="#">Section 3.4.3.118</a>
464h	PPSE4MSTID_H	PPSE4MSTID_H	<a href="#">Section 3.4.3.119</a>
468h	PPSE5MSTID_L	PPSE5MSTID_L	<a href="#">Section 3.4.3.120</a>
46Ch	PPSE5MSTID_H	PPSE5MSTID_H	<a href="#">Section 3.4.3.121</a>
470h	PPSE6MSTID_L	PPSE6MSTID_L	<a href="#">Section 3.4.3.122</a>
474h	PPSE6MSTID_H	PPSE6MSTID_H	<a href="#">Section 3.4.3.123</a>
478h	PPSE7MSTID_L	PPSE7MSTID_L	<a href="#">Section 3.4.3.124</a>
47Ch	PPSE7MSTID_H	PPSE7MSTID_H	<a href="#">Section 3.4.3.125</a>
480h	PPSE8MSTID_L	PPSE8MSTID_L	<a href="#">Section 3.4.3.126</a>
484h	PPSE8MSTID_H	PPSE8MSTID_H	<a href="#">Section 3.4.3.127</a>
488h	PPSE9MSTID_L	PPSE9MSTID_L	<a href="#">Section 3.4.3.128</a>
48Ch	PPSE9MSTID_H	PPSE9MSTID_H	<a href="#">Section 3.4.3.129</a>
490h	PPSE10MSTID_L	PPSE10MSTID_L	<a href="#">Section 3.4.3.130</a>

**Table 3-437. DSS\_PCR Registers (continued)**

Offset	Acronym	Register Name	Section
494h	PPSE10MSTID_H	PPSE10MSTID_H	<a href="#">Section 3.4.3.131</a>
498h	PPSE11MSTID_L	PPSE11MSTID_L	<a href="#">Section 3.4.3.132</a>
49Ch	PPSE11MSTID_H	PPSE11MSTID_H	<a href="#">Section 3.4.3.133</a>
4A0h	PPSE12MSTID_L	PPSE12MSTID_L	<a href="#">Section 3.4.3.134</a>
4A4h	PPSE12MSTID_H	PPSE12MSTID_H	<a href="#">Section 3.4.3.135</a>
4A8h	PPSE13MSTID_L	PPSE13MSTID_L	<a href="#">Section 3.4.3.136</a>
4ACh	PPSE13MSTID_H	PPSE13MSTID_H	<a href="#">Section 3.4.3.137</a>
4B0h	PPSE14MSTID_L	PPSE14MSTID_L	<a href="#">Section 3.4.3.138</a>
4B4h	PPSE14MSTID_H	PPSE14MSTID_H	<a href="#">Section 3.4.3.139</a>
4B8h	PPSE15MSTID_L	PPSE15MSTID_L	<a href="#">Section 3.4.3.140</a>
4BCh	PPSE15MSTID_H	PPSE15MSTID_H	<a href="#">Section 3.4.3.141</a>
4C0h	PPSE16MSTID_L	PPSE16MSTID_L	<a href="#">Section 3.4.3.142</a>
4C4h	PPSE16MSTID_H	PPSE16MSTID_H	<a href="#">Section 3.4.3.143</a>
4C8h	PPSE17MSTID_L	PPSE17MSTID_L	<a href="#">Section 3.4.3.144</a>
4CCh	PPSE17MSTID_H	PPSE17MSTID_H	<a href="#">Section 3.4.3.145</a>
4D0h	PPSE18MSTID_L	PPSE18MSTID_L	<a href="#">Section 3.4.3.146</a>
4D4h	PPSE18MSTID_H	PPSE18MSTID_H	<a href="#">Section 3.4.3.147</a>
4D8h	PPSE19MSTID_L	PPSE19MSTID_L	<a href="#">Section 3.4.3.148</a>
4DCh	PPSE19MSTID_H	PPSE19MSTID_H	<a href="#">Section 3.4.3.149</a>
4E0h	PPSE20MSTID_L	PPSE20MSTID_L	<a href="#">Section 3.4.3.150</a>
4E4h	PPSE20MSTID_H	PPSE20MSTID_H	<a href="#">Section 3.4.3.151</a>
4E8h	PPSE21MSTID_L	PPSE21MSTID_L	<a href="#">Section 3.4.3.152</a>
4ECh	PPSE21MSTID_H	PPSE21MSTID_H	<a href="#">Section 3.4.3.153</a>
4F0h	PPSE22MSTID_L	PPSE22MSTID_L	<a href="#">Section 3.4.3.154</a>
4F4h	PPSE22MSTID_H	PPSE22MSTID_H	<a href="#">Section 3.4.3.155</a>
4F8h	PPSE23MSTID_L	PPSE23MSTID_L	<a href="#">Section 3.4.3.156</a>
4FCh	PPSE23MSTID_H	PPSE23MSTID_H	<a href="#">Section 3.4.3.157</a>
500h	PPSE24MSTID_L	PPSE24MSTID_L	<a href="#">Section 3.4.3.158</a>
504h	PPSE24MSTID_H	PPSE24MSTID_H	<a href="#">Section 3.4.3.159</a>
508h	PPSE25MSTID_L	PPSE25MSTID_L	<a href="#">Section 3.4.3.160</a>
50Ch	PPSE25MSTID_H	PPSE25MSTID_H	<a href="#">Section 3.4.3.161</a>
510h	PPSE26MSTID_L	PPSE26MSTID_L	<a href="#">Section 3.4.3.162</a>
514h	PPSE26MSTID_H	PPSE26MSTID_H	<a href="#">Section 3.4.3.163</a>
518h	PPSE27MSTID_L	PPSE27MSTID_L	<a href="#">Section 3.4.3.164</a>
51Ch	PPSE27MSTID_H	PPSE27MSTID_H	<a href="#">Section 3.4.3.165</a>
520h	PPSE28MSTID_L	PPSE28MSTID_L	<a href="#">Section 3.4.3.166</a>
524h	PPSE28MSTID_H	PPSE28MSTID_H	<a href="#">Section 3.4.3.167</a>
528h	PPSE29MSTID_L	PPSE29MSTID_L	<a href="#">Section 3.4.3.168</a>
52Ch	PPSE29MSTID_H	PPSE29MSTID_H	<a href="#">Section 3.4.3.169</a>
530h	PPSE30MSTID_L	PPSE30MSTID_L	<a href="#">Section 3.4.3.170</a>
534h	PPSE30MSTID_H	PPSE30MSTID_H	<a href="#">Section 3.4.3.171</a>
538h	PPSE31MSTID_L	PPSE31MSTID_L	<a href="#">Section 3.4.3.172</a>
53Ch	PPSE31MSTID_H	PPSE31MSTID_H	<a href="#">Section 3.4.3.173</a>
540h	PCS0MSTID	PCS0MSTID	<a href="#">Section 3.4.3.174</a>
544h	PCS1MSTID	PCS1MSTID	<a href="#">Section 3.4.3.175</a>



**Table 3-437. DSS\_PCR Registers (continued)**

Offset	Acronym	Register Name	Section
548h	PCS2MSTID	PCS2MSTID	<a href="#">Section 3.4.3.176</a>
54Ch	PCS3MSTID	PCS3MSTID	<a href="#">Section 3.4.3.177</a>
550h	PCS4MSTID	PCS4MSTID	<a href="#">Section 3.4.3.178</a>
554h	PCS5MSTID	PCS5MSTID	<a href="#">Section 3.4.3.179</a>
558h	PCS6MSTID	PCS6MSTID	<a href="#">Section 3.4.3.180</a>
55Ch	PCS7MSTID	PCS7MSTID	<a href="#">Section 3.4.3.181</a>
560h	PCS8MSTID	PCS8MSTID	<a href="#">Section 3.4.3.182</a>
564h	PCS9MSTID	PCS9MSTID	<a href="#">Section 3.4.3.183</a>
568h	PCS10MSTID	PCS10MSTID	<a href="#">Section 3.4.3.184</a>
56Ch	PCS11MSTID	PCS11MSTID	<a href="#">Section 3.4.3.185</a>
570h	PCS12MSTID	PCS12MSTID	<a href="#">Section 3.4.3.186</a>
574h	PCS13MSTID	PCS13MSTID	<a href="#">Section 3.4.3.187</a>
578h	PCS14MSTID	PCS14MSTID	<a href="#">Section 3.4.3.188</a>
57Ch	PCS15MSTID	PCS15MSTID	<a href="#">Section 3.4.3.189</a>
580h	PCS16MSTID	PCS16MSTID	<a href="#">Section 3.4.3.190</a>
584h	PCS17MSTID	PCS17MSTID	<a href="#">Section 3.4.3.191</a>
588h	PCS18MSTID	PCS18MSTID	<a href="#">Section 3.4.3.192</a>
58Ch	PCS19MSTID	PCS19MSTID	<a href="#">Section 3.4.3.193</a>
590h	PCS20MSTID	PCS20MSTID	<a href="#">Section 3.4.3.194</a>
594h	PCS21MSTID	PCS21MSTID	<a href="#">Section 3.4.3.195</a>
598h	PCS22MSTID	PCS22MSTID	<a href="#">Section 3.4.3.196</a>
59Ch	PCS23MSTID	PCS23MSTID	<a href="#">Section 3.4.3.197</a>
5A0h	PCS24MSTID	PCS24MSTID	<a href="#">Section 3.4.3.198</a>
5A4h	PCS25MSTID	PCS25MSTID	<a href="#">Section 3.4.3.199</a>
5A8h	PCS26MSTID	PCS26MSTID	<a href="#">Section 3.4.3.200</a>
5ACh	PCS27MSTID	PCS27MSTID	<a href="#">Section 3.4.3.201</a>
5B0h	PCS28MSTID	PCS28MSTID	<a href="#">Section 3.4.3.202</a>
5B4h	PCS29MSTID	PCS29MSTID	<a href="#">Section 3.4.3.203</a>
5B8h	PCS30MSTID	PCS30MSTID	<a href="#">Section 3.4.3.204</a>
5BCh	PCS31MSTID	PCS31MSTID	<a href="#">Section 3.4.3.205</a>
5C0h	PPCS0MSTID	PPCS0MSTID	<a href="#">Section 3.4.3.206</a>
5C4h	PPCS1MSTID	PPCS1MSTID	<a href="#">Section 3.4.3.207</a>
5C8h	PPCS2MSTID	PPCS2MSTID	<a href="#">Section 3.4.3.208</a>
5CCh	PPCS3MSTID	PPCS3MSTID	<a href="#">Section 3.4.3.209</a>
5D0h	PPCS4MSTID	PPCS4MSTID	<a href="#">Section 3.4.3.210</a>
5D4h	PPCS5MSTID	PPCS5MSTID	<a href="#">Section 3.4.3.211</a>
5D8h	PPCS6MSTID	PPCS6MSTID	<a href="#">Section 3.4.3.212</a>
5DCh	PPCS7MSTID	PPCS7MSTID	<a href="#">Section 3.4.3.213</a>
5E0h	PCREXTMSTID	PCREXTMSTID	<a href="#">Section 3.4.3.214</a>

Complex bit access types are encoded to fit into small table cells. [Table 3-438](#) shows the codes that are used for access types in this section.

**Table 3-438. DSS\_PCR Access Type Codes**

Access Type	Code	Description
Read Type		

**Table 3-438. DSS\_PCR Access Type Codes  
(continued)**

Access Type	Code	Description
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 3.4.3.1 PMPROTSET0 Register (Offset = 0h) [Reset = 00000000h]

PMPROTSET0 is shown in [Table 3-439](#).

Return to the [Summary Table](#).

Set-only register to protect PCS frames 0 to 31

**Table 3-439. PMPROTSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
30	PCS30_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
29	PCS29_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
28	PCS28_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
27	PCS27_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-439. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
26	PCS26_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
25	PCS25_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
24	PCS24_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
23	PCS23_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
22	PCS22_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
21	PCS21_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-439. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
20	PCS20_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
19	PCS19_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
18	PCS18_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
17	PCS17_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
16	PCS16_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
15	PCS15_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-439. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	PCS14_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
13	PCS13_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
12	PCS12_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
11	PCS11_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
10	PCS10_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
9	PCS9_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.

**Table 3-439. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS8_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
7	PCS7_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
6	PCS6_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
5	PCS5_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
4	PCS4_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
3	PCS3_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.



**Table 3-439. PMPROTSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
1	PCS1_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.
0	PCS0_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET0 and PMPROTCLR0 registers 0 = Has no effect Only those bits which have a slave at the corresponding bit position are implemented. Hence, the size of this register is device dependent. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.2 PMPROTSET1 Register (Offset = 4h) [Reset = 00000000h]

PMPROTSET1 is shown in [Table 3-440](#).

Return to the [Summary Table](#).

Set-only register to protect PCS frames 32 to 63

**Table 3-440. PMPROTSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
30	PCS62_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
29	PCS61_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
28	PCS60_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
27	PCS59_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
26	PCS58_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
25	PCS57_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-440. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS56_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
23	PCS55_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
22	PCS54_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
21	PCS53_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
20	PCS52_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
19	PCS51_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
18	PCS50_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
17	PCS49_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-440. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS48_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
15	PCS47_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
14	PCS46_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
13	PCS45_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
12	PCS44_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
11	PCS43_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
10	PCS42_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
9	PCS41_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-440. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS40_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
7	PCS39_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
6	PCS38_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
5	PCS37_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
4	PCS36_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
3	PCS35_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
2	PCS34_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect
1	PCS33_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

**Table 3-440. PMPROTSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS32_PROT_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PMPROTSET1 and PMPROTCLR1 registers 0 = Has no effect

### 3.4.3.3 PMPROTCLR0 Register (Offset = 10h) [Reset = 00000000h]

PMPROTCLR0 is shown in [Table 3-441](#).

Return to the [Summary Table](#).

Clear-only register to protect PCS frames 0 to 31

**Table 3-441. PMPROTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
30	PCS30_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
29	PCS29_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
28	PCS28_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
27	PCS27_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
26	PCS26_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
25	PCS25_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

**Table 3-441. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS24_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
23	PCS23_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
22	PCS22_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
21	PCS21_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
20	PCS20_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
19	PCS19_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
18	PCS18_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
17	PCS17_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect



**Table 3-441. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS16_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
15	PCS15_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
14	PCS14_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
13	PCS13_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
12	PCS12_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
11	PCS11_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
10	PCS10_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
9	PCS9_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect



**Table 3-441. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS8_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
7	PCS7_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
6	PCS6_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
5	PCS5_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
4	PCS4_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
3	PCS3_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
2	PCS2_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect
1	PCS1_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

**Table 3-441. PMPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS0_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR0 and PMPROTSET0 registers 0 = Has no effect

### 3.4.3.4 PMPROTCLR1 Register (Offset = 14h) [Reset = 00000000h]

PMPROTCLR1 is shown in [Table 3-442](#).

Return to the [Summary Table](#).

Clear-only register to protect PCS frames 32 to 63

**Table 3-442. PMPROTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
30	PCS62_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
29	PCS61_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
28	PCS60_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
27	PCS59_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
26	PCS58_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
25	PCS57_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-442. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PCS56_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
23	PCS55_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
22	PCS54_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
21	PCS53_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
20	PCS52_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
19	PCS51_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
18	PCS50_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
17	PCS49_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-442. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PCS48_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
15	PCS47_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
14	PCS46_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
13	PCS45_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
12	PCS44_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
11	PCS43_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
10	PCS42_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
9	PCS41_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-442. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PCS40_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
7	PCS39_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
6	PCS38_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
5	PCS37_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
4	PCS36_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
3	PCS35_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
2	PCS34_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect
1	PCS33_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect

**Table 3-442. PMPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PCS32_PROT_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory frame can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PMPROTCLR1 and PMPROTSET1 registers 0 = Has no effect



### 3.4.3.5 PPROTSET\_0 Register (Offset = 20h) [Reset = 00000000h]

PPROTSET\_0 is shown in [Table 3-443](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS0 to PS7

**Table 3-443. PPROTSET\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



**Table 3-443. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-443. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-443. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-443. PPROTSET\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

### 3.4.3.6 PPROTSET\_1 Register (Offset = 24h) [Reset = 00000000h]

PPROTSET\_1 is shown in [Table 3-444](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS8 to PS15

**Table 3-444. PPROTSET\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-444. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-444. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect



**Table 3-444. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect



**Table 3-444. PPROTSET\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PROT_SET	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

### 3.4.3.7 PPROTSET\_2 Register (Offset = 28h) [Reset = 00000000h]

PPROTSET\_2 is shown in [Table 3-445](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS16 to PS23

**Table 3-445. PPROTSET\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-445. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-445. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-445. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-445. PPROTSET\_2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

### 3.4.3.8 PPROTSET\_3 Register (Offset = 2Ch) [Reset = 00000000h]

PPROTSET\_3 is shown in [Table 3-446](#).

Return to the [Summary Table](#).

Set-only register to protect the 32 quadrants of PS24 to PS31

**Table 3-446. PPROTSET\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



**Table 3-446. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



**Table 3-446. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-446. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-446. PPROTSET\_3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PROT_SE T	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Sets the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

### 3.4.3.9 PPROTCLR0 Register (Offset = 40h) [Reset = 00000000h]

PPROTCLR0 is shown in [Table 3-447](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS0 to PS7

**Table 3-447. PPROTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-447. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-447. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect

**Table 3-447. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



**Table 3-447. PPROTCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET0 and PPROTCLR0 registers 0 = Has no effect



### 3.4.3.10 PPROTCLR1 Register (Offset = 44h) [Reset = 00000000h]

PPROTCLR1 is shown in [Table 3-448](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS8 to PS15

**Table 3-448. PPROTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-448. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-448. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-448. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

**Table 3-448. PPROTCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET1 and PPROTCLR1 registers 0 = Has no effect

### 3.4.3.11 PPROTCLR2 Register (Offset = 48h) [Reset = 00000000h]

PPROTCLR2 is shown in [Table 3-449](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS16 to PS23

**Table 3-449. PPROTCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-449. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect



**Table 3-449. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect



**Table 3-449. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

**Table 3-449. PPROTCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET2 and PPROTCLR2 registers 0 = Has no effect

### 3.4.3.12 PPROTCLR3 Register (Offset = 4Ch) [Reset = 00000000h]

PPROTCLR3 is shown in [Table 3-450](#).

Return to the [Summary Table](#).

Clear-only register to protect the 32 quadrants of PS24 to PS31

**Table 3-450. PPROTCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-450. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-450. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-450. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect

**Table 3-450. PPROTCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PROT_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The quadrant 'm' of the peripheral frame 'n' can be written to only in privileged mode but can be read in both user and privileged modes. 0 = The corresponding peripheral memory frame can be written to and read from in both user and privileged modes. Writable only in privileged mode 1 = Clears the corresponding bit in PPROTSET3 and PPROTCLR3 registers 0 = Has no effect



### 3.4.3.13 PCSPWRDWNSET0 Register (Offset = 60h) [Reset = 00000000h]

PCSPWRDWNSET0 is shown in [Table 3-451](#).

Return to the [Summary Table](#).

Set-only register to powerdown independent (non-shared) PCS frames 0 to 31

**Table 3-451. PCSPWRDWNSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-451. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS22_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-451. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS12_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-451. PCSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.3.14 PCSPWRDWNSET1 Register (Offset = 64h) [Reset = 00000000h]

PCSPWRDWNSET1 is shown in [Table 3-452](#).

Return to the [Summary Table](#).

Set-only register to powerdown independent (non-shared) PCS frames 32 to 63

**Table 3-452. PCSPWRDWNSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-452. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS54_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-452. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS44_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-452. PCSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS34_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_SET	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Sets the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect



### 3.4.3.15 PCSPWRDWNCLR0 Register (Offset = 70h) [Reset = 00000000h]

PCSPWRDWNCLR0 is shown in [Table 3-453](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 0 to 31

**Table 3-453. PCSPWRDWNCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS31_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
30	PCS30_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
29	PCS29_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
28	PCS28_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
27	PCS27_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
26	PCS26_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
25	PCS25_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
24	PCS24_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
23	PCS23_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-453. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS22_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
21	PCS21_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
20	PCS20_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
19	PCS19_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
18	PCS18_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
17	PCS17_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
16	PCS16_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
15	PCS15_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
14	PCS14_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
13	PCS13_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-453. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS12_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
11	PCS11_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
10	PCS10_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
9	PCS9_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
8	PCS8_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
7	PCS7_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
6	PCS6_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
5	PCS5_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
4	PCS4_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
3	PCS3_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-453. PCSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS2_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
1	PCS1_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect
0	PCS0_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET0 and PCSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.3.16 PCSPWRDWNCLR1 Register (Offset = 74h) [Reset = 00000000h]

PCSPWRDWNCLR1 is shown in [Table 3-454](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of independent (non-shared) PCS frames 32 to 63

**Table 3-454. PCSPWRDWNCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PCS63_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
30	PCS62_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
29	PCS61_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
28	PCS60_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
27	PCS59_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
26	PCS58_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
25	PCS57_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
24	PCS56_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
23	PCS55_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-454. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
22	PCS54_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
21	PCS53_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
20	PCS52_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
19	PCS51_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
18	PCS50_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
17	PCS49_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
16	PCS48_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
15	PCS47_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
14	PCS46_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
13	PCS45_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-454. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
12	PCS44_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
11	PCS43_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
10	PCS42_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
9	PCS41_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
8	PCS40_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
7	PCS39_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
6	PCS38_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
5	PCS37_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
4	PCS36_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
3	PCS35_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-454. PCSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	PCS34_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
1	PCS33_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect
0	PCS32_PWRDWN_CLR	R/W	0h	Readable in user and privileged modes 1 = The corresponding peripheral memory clock needs to be powered down. 0 = The corresponding peripheral memory clock is not to be powered down. Writable only in privileged mode 1 = Clears the corresponding bit in PCSPWRDWNSET1 and PCSPWRDWNCLR1 registers 0 = Has no effect



### 3.4.3.17 PSPWRDWNSET0 Register (Offset = 80h) [Reset = 00000000h]

PSPWRDWNSET0 is shown in [Table 3-455](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS0 to PS7

**Table 3-455. PSPWRDWNSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect



**Table 3-455. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-455. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-455. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-455. PSPWRDWNSET0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

### 3.4.3.18 PSPWRDWNSET1 Register (Offset = 84h) [Reset = 00000000h]

PSPWRDWNSET1 is shown in [Table 3-456](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS8 to PS15

**Table 3-456. PSPWRDWNSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-456. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-456. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect



**Table 3-456. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect



**Table 3-456. PSPWRDWNSET1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET1 and PSPWRDWNCLR1 registers 0 = Has no effect

### 3.4.3.19 PSPWRDWNSET2 Register (Offset = 88h) [Reset = 00000000h]

PSPWRDWNSET2 is shown in [Table 3-457](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS16 to PS23

**Table 3-457. PSPWRDWNSET2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-457. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-457. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-457. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-457. PSPWRDWNSET2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

### 3.4.3.20 PSPWRDWNSET3 Register (Offset = 8Ch) [Reset = 00000000h]

PSPWRDWNSET3 is shown in [Table 3-458](#).

Return to the [Summary Table](#).

Set-only register to powerdown the applicable peripherals in the 32 quadrants of PS24 to PS31

**Table 3-458. PSPWRDWNSET3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
30	PS31_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-458. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS30_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
23	PS29_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect



**Table 3-458. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS28_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
15	PS27_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-458. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS26_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
7	PS25_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-458. PSPWRDWNSET3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS24_QUAD0_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Sets the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

### 3.4.3.21 PSPWRDWNCLR0 Register (Offset = A0h) [Reset = 00000000h]

PSPWRDWNCLR0 is shown in [Table 3-459](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS0 to PS7

**Table 3-459. PSPWRDWNCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS7_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
30	PS7_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
29	PS7_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
28	PS7_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
27	PS6_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
26	PS6_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
25	PS6_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-459. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS6_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
23	PS5_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
22	PS5_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
21	PS5_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
20	PS5_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
19	PS4_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
18	PS4_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
17	PS4_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-459. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS4_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
15	PS3_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
14	PS3_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
13	PS3_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
12	PS3_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
11	PS2_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
10	PS2_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
9	PS2_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-459. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS2_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
7	PS1_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
6	PS1_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
5	PS1_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
4	PS1_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
3	PS0_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
2	PS0_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect
1	PS0_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect

**Table 3-459. PSPWRDWNCLR0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS0_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET0 and PSPWRDWNCLR0 registers 0 = Has no effect



### 3.4.3.22 PSPWRDWNCLR1 Register (Offset = A4h) [Reset = 00000000h]

PSPWRDWNCLR1 is shown in [Table 3-460](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS8 to PS15

**Table 3-460. PSPWRDWNCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS15_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
30	PS15_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
29	PS15_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
28	PS15_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
27	PS14_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
26	PS14_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
25	PS14_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-460. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS14_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
23	PS13_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
22	PS13_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
21	PS13_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
20	PS13_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
19	PS12_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
18	PS12_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
17	PS12_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-460. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS12_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
15	PS11_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
14	PS11_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
13	PS11_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
12	PS11_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
11	PS10_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
10	PS10_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
9	PS10_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-460. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS10_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
7	PS9_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
6	PS9_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
5	PS9_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
4	PS9_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
3	PS8_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
2	PS8_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect
1	PS8_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

**Table 3-460. PSPWRDWNCLR1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS8_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNCLR1 and PSPWRDWNCLR1 registers 0 = Has no effect

### 3.4.3.23 PSPWRDWNCLR2 Register (Offset = A8h) [Reset = 00000000h]

PSPWRDWNCLR2 is shown in [Table 3-461](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS16 to PS23

**Table 3-461. PSPWRDWNCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS23_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
30	PS23_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
29	PS23_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
28	PS23_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
27	PS22_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
26	PS22_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
25	PS22_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-461. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
24	PS22_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
23	PS21_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
22	PS21_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
21	PS21_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
20	PS21_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
19	PS20_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
18	PS20_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
17	PS20_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect



**Table 3-461. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
16	PS20_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
15	PS19_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
14	PS19_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
13	PS19_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
12	PS19_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
11	PS18_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
10	PS18_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
9	PS18_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect



**Table 3-461. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	PS18_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
7	PS17_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
6	PS17_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
5	PS17_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
4	PS17_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
3	PS16_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
2	PS16_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect
1	PS16_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

**Table 3-461. PSPWRDWNCLR2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	PS16_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET2 and PSPWRDWNCLR2 registers 0 = Has no effect

### 3.4.3.24 PSPWRDWNCLR3 Register (Offset = ACh) [Reset = 00000000h]

PSPWRDWNCLR3 is shown in [Table 3-462](#).

Return to the [Summary Table](#).

Clear-only register to deassert powerdown bits of the applicable peripherals in the 32 quadrants of PS24 to PS31

**Table 3-462. PSPWRDWNCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PS31_QUAD3_PWRDWN_CLR	R/W	0h	
30	PS31_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
29	PS31_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
28	PS31_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
27	PS30_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
26	PS30_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
25	PS30_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
24	PS30_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-462. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
23	PS29_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
22	PS29_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
21	PS29_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
20	PS29_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
19	PS28_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
18	PS28_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
17	PS28_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
16	PS28_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-462. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
15	PS27_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
14	PS27_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
13	PS27_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
12	PS27_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
11	PS26_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
10	PS26_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
9	PS26_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
8	PS26_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

**Table 3-462. PSPWRDWNCLR3 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7	PS25_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
6	PS25_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
5	PS25_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
4	PS25_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
3	PS24_QUAD3_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
2	PS24_QUAD2_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
1	PS24_QUAD1_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect
0	PS24_QUAD0_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered down. 0 = The clock to the peripheral starting at quadrant 'm' of the peripheral frame 'n' needs to be powered up. Writable only in privileged mode 1 = Clears the corresponding bit in PSPWRDWNSET3 and PSPWRDWNCLR3 registers 0 = Has no effect

### 3.4.3.25 PDPWRDWNSET Register (Offset = C0h) [Reset = 00000000h]

PDPWRDWNSET is shown in [Table 3-463](#).

Return to the [Summary Table](#).

Set-only register to powerdown the debug frame

**Table 3-463. PDPWRDWNSET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_SET	R/W	0h	Readable in both user and privileged modes. 1 = Clock to the debug frame needs to be powered down. 0 = Clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get set in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect

### 3.4.3.26 PDPWRDWNCLR Register (Offset = C4h) [Reset = 00000000h]

PDPWRDWNCLR is shown in [Table 3-464](#).

Return to the [Summary Table](#).

Clear-only register to deassert the debug frame's powerdown bit

**Table 3-464. PDPWRDWNCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	PD_PWRDWN_CLR	R/W	0h	Readable in both user and privileged modes. 1 = The clock to the debug frame needs to be powered down. 0 = The clock to the debug frame needs to be powered up. Writable only in privileged mode 1 = Bit 0 when written 1, will get cleared in both PDPWRDWNSET and PDPWRDWNCLR registers. The other bits are not affected. 0 = Has no effect



### 3.4.3.27 MSTIDWRENA Register (Offset = 200h) [Reset = 00000000h]

MSTIDWRENA is shown in [Table 3-465](#).

Return to the [Summary Table](#).

MasterID Protection Write Enable Register

**Table 3-465. MSTIDWRENA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTIDREG_WRENA	R/W	0h	Readable in both user and privileged modes. 1010 = All master-id registers are unlocked and available for write. others = Writes to all master-id registers are locked. Writable only in privileged mode 1010 = Writes to master-id registers are unlocked. others = Writes to master-id registers are locked.

### 3.4.3.28 MSTIDENA Register (Offset = 204h) [Reset = 00000000h]

MSTIDENA is shown in [Table 3-466](#).

Return to the [Summary Table](#).

MasterID Protection Enable Register

**Table 3-466. MSTIDENA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	MSTID_CHK_EN	R/W	0h	Readable in both user and privileged modes. Writable only in privileged mode 1010 = Enable the master-id feature check. others = Master-id check is disabled.

### 3.4.3.29 MSTIDDIAGCTRL Register (Offset = 208h) [Reset = 00000000h]

MSTIDDIAGCTRL is shown in [Table 3-467](#).

Return to the [Summary Table](#).

MasterID Diagnostic Control Register

**Table 3-467. MSTIDDIAGCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	Reserved
11-8	DIAG_CMP_VALUE	R/W	0h	MasterID diagnostic mode control register bits; 4-bit data which is compared with the master-id register of all defined frames during diagnostic mode. Any error in compare logic is indicated through AERROR output from PCR. Readable in both user and privileged modes. Reads the programmed value in diagnostic compare value field. Writable only in privileged mode
7-4	RESERVED	R	0h	Reserved
3-0	DIAG_MODE_EN	R/W	0h	MasterID compare logic diagnostic mode enable bits; 4-bit key for enabling the master-id registers compare logic. Readable in both user and privileged modes. Writable only in privileged mode 1010 = Master-id compare diagnostic mode is enabled. others = Master-id compare diagnostic mode is disabled.

### 3.4.3.30 PS0MSTID\_L Register (Offset = 300h) [Reset = 00000000h]

PS0MSTID\_L is shown in [Table 3-468](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register0\_L

**Table 3-468. PS0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.31 PS0MSTID\_H Register (Offset = 304h) [Reset = 00000000h]

PS0MSTID\_H is shown in [Table 3-469](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register0\_H

**Table 3-469. PS0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.32 PS1MSTID\_L Register (Offset = 308h) [Reset = 00000000h]

PS1MSTID\_L is shown in [Table 3-470](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register1\_L

**Table 3-470. PS1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.33 PS1MSTID\_H Register (Offset = 30Ch) [Reset = 00000000h]

PS1MSTID\_H is shown in [Table 3-471](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register1\_H

**Table 3-471. PS1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.34 PS2MSTID\_L Register (Offset = 310h) [Reset = 00000000h]

PS2MSTID\_L is shown in [Table 3-472](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register2\_L

**Table 3-472. PS2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.35 PS2MSTID\_H Register (Offset = 314h) [Reset = 00000000h]

PS2MSTID\_H is shown in [Table 3-473](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register2\_H

**Table 3-473. PS2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.36 PS3MSTID\_L Register (Offset = 318h) [Reset = 00000000h]

PS3MSTID\_L is shown in [Table 3-474](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register3\_L

**Table 3-474. PS3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.37 PS3MSTID\_H Register (Offset = 31Ch) [Reset = 00000000h]

PS3MSTID\_H is shown in [Table 3-475](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register3\_H

**Table 3-475. PS3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.38 PS4MSTID\_L Register (Offset = 320h) [Reset = 00000000h]

PS4MSTID\_L is shown in [Table 3-476](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register4\_L

**Table 3-476. PS4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.39 PS4MSTID\_H Register (Offset = 324h) [Reset = 00000000h]

PS4MSTID\_H is shown in [Table 3-477](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register4\_H

**Table 3-477. PS4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.40 PS5MSTID\_L Register (Offset = 328h) [Reset = 00000000h]

PS5MSTID\_L is shown in [Table 3-478](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register5\_L

**Table 3-478. PS5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.41 PS5MSTID\_H Register (Offset = 32Ch) [Reset = 00000000h]

PS5MSTID\_H is shown in [Table 3-479](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register5\_H

**Table 3-479. PS5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.42 PS6MSTID\_L Register (Offset = 330h) [Reset = 00000000h]

PS6MSTID\_L is shown in [Table 3-480](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register6\_L

**Table 3-480. PS6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.43 PS6MSTID\_H Register (Offset = 334h) [Reset = 00000000h]

PS6MSTID\_H is shown in [Table 3-481](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register6\_H

**Table 3-481. PS6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.44 PS7MSTID\_L Register (Offset = 338h) [Reset = 00000000h]

PS7MSTID\_L is shown in [Table 3-482](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register7\_L

**Table 3-482. PS7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.45 PS7MSTID\_H Register (Offset = 33Ch) [Reset = 00000000h]

PS7MSTID\_H is shown in [Table 3-483](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register7\_H

**Table 3-483. PS7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.46 PS8MSTID\_L Register (Offset = 340h) [Reset = 00000000h]

PS8MSTID\_L is shown in [Table 3-484](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register8\_L

**Table 3-484. PS8MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.47 PS8MSTID\_H Register (Offset = 344h) [Reset = 00000000h]

PS8MSTID\_H is shown in [Table 3-485](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register8\_H

**Table 3-485. PS8MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.48 PS9MSTID\_L Register (Offset = 348h) [Reset = 00000000h]

PS9MSTID\_L is shown in [Table 3-486](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register9\_L

**Table 3-486. PS9MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.49 PS9MSTID\_H Register (Offset = 34Ch) [Reset = 00000000h]

PS9MSTID\_H is shown in [Table 3-487](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register9\_H

**Table 3-487. PS9MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.50 PS10MSTID\_L Register (Offset = 350h) [Reset = 00000000h]

PS10MSTID\_L is shown in [Table 3-488](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register10\_L

**Table 3-488. PS10MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.51 PS10MSTID\_H Register (Offset = 354h) [Reset = 00000000h]

PS10MSTID\_H is shown in [Table 3-489](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register10\_H

**Table 3-489. PS10MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.52 PS11MSTID\_L Register (Offset = 358h) [Reset = 00000000h]

PS11MSTID\_L is shown in [Table 3-490](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register11\_L

**Table 3-490. PS11MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.53 PS11MSTID\_H Register (Offset = 35Ch) [Reset = 00000000h]

PS11MSTID\_H is shown in [Table 3-491](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register11\_H

**Table 3-491. PS11MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.54 PS12MSTID\_L Register (Offset = 360h) [Reset = 00000000h]

PS12MSTID\_L is shown in [Table 3-492](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register12\_L

**Table 3-492. PS12MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.55 PS12MSTID\_H Register (Offset = 364h) [Reset = 00000000h]

PS12MSTID\_H is shown in [Table 3-493](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register12\_H

**Table 3-493. PS12MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.56 PS13MSTID\_L Register (Offset = 368h) [Reset = 00000000h]

PS13MSTID\_L is shown in [Table 3-494](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register13\_L

**Table 3-494. PS13MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.57 PS13MSTID\_H Register (Offset = 36Ch) [Reset = 00000000h]

PS13MSTID\_H is shown in [Table 3-495](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register13\_H

**Table 3-495. PS13MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.58 PS14MSTID\_L Register (Offset = 370h) [Reset = 00000000h]

PS14MSTID\_L is shown in [Table 3-496](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register14\_L

**Table 3-496. PS14MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.59 PS14MSTID\_H Register (Offset = 374h) [Reset = 00000000h]

PS14MSTID\_H is shown in [Table 3-497](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register14\_H

**Table 3-497. PS14MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.60 PS15MSTID\_L Register (Offset = 378h) [Reset = 00000000h]

PS15MSTID\_L is shown in [Table 3-498](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register15\_L

**Table 3-498. PS15MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.61 PS15MSTID\_H Register (Offset = 37Ch) [Reset = 00000000h]

PS15MSTID\_H is shown in [Table 3-499](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register15\_H

**Table 3-499. PS15MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.62 PS16MSTID\_L Register (Offset = 380h) [Reset = 00000000h]

PS16MSTID\_L is shown in [Table 3-500](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register16\_L

**Table 3-500. PS16MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.63 PS16MSTID\_H Register (Offset = 384h) [Reset = 00000000h]

PS16MSTID\_H is shown in [Table 3-501](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register16\_H

**Table 3-501. PS16MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.64 PS17MSTID\_L Register (Offset = 388h) [Reset = 00000000h]

PS17MSTID\_L is shown in [Table 3-502](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register17\_L

**Table 3-502. PS17MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.65 PS17MSTID\_H Register (Offset = 38Ch) [Reset = 00000000h]

PS17MSTID\_H is shown in [Table 3-503](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register17\_H

**Table 3-503. PS17MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.66 PS18MSTID\_L Register (Offset = 390h) [Reset = 00000000h]

PS18MSTID\_L is shown in [Table 3-504](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register18\_L

**Table 3-504. PS18MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.67 PS18MSTID\_H Register (Offset = 394h) [Reset = 00000000h]

PS18MSTID\_H is shown in [Table 3-505](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register18\_H

**Table 3-505. PS18MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.68 PS19MSTID\_L Register (Offset = 398h) [Reset = 00000000h]

PS19MSTID\_L is shown in [Table 3-506](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register19\_L

**Table 3-506. PS19MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.69 PS19MSTID\_H Register (Offset = 39Ch) [Reset = 00000000h]

PS19MSTID\_H is shown in [Table 3-507](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register19\_H

**Table 3-507. PS19MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.70 PS20MSTID\_L Register (Offset = 3A0h) [Reset = 00000000h]

PS20MSTID\_L is shown in [Table 3-508](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register20\_L

**Table 3-508. PS20MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.71 PS20MSTID\_H Register (Offset = 3A4h) [Reset = 00000000h]

PS20MSTID\_H is shown in [Table 3-509](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register20\_H

**Table 3-509. PS20MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.72 PS21MSTID\_L Register (Offset = 3A8h) [Reset = 00000000h]

PS21MSTID\_L is shown in [Table 3-510](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register21\_L

**Table 3-510. PS21MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.73 PS21MSTID\_H Register (Offset = 3ACh) [Reset = 00000000h]

PS21MSTID\_H is shown in [Table 3-511](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register21\_H

**Table 3-511. PS21MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.74 PS22MSTID\_L Register (Offset = 3B0h) [Reset = 00000000h]

PS22MSTID\_L is shown in [Table 3-512](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register22\_L

**Table 3-512. PS22MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.75 PS22MSTID\_H Register (Offset = 3B4h) [Reset = 00000000h]

PS22MSTID\_H is shown in [Table 3-513](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register22\_H

**Table 3-513. PS22MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.76 PS23MSTID\_L Register (Offset = 3B8h) [Reset = 00000000h]

PS23MSTID\_L is shown in [Table 3-514](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register23\_L

**Table 3-514. PS23MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.77 PS23MSTID\_H Register (Offset = 3BCh) [Reset = 00000000h]

PS23MSTID\_H is shown in [Table 3-515](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register23\_H

**Table 3-515. PS23MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.78 PS24MSTID\_L Register (Offset = 3C0h) [Reset = 00000000h]

PS24MSTID\_L is shown in [Table 3-516](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register24\_L

**Table 3-516. PS24MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.79 PS24MSTID\_H Register (Offset = 3C4h) [Reset = 00000000h]

PS24MSTID\_H is shown in [Table 3-517](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register24\_H

**Table 3-517. PS24MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.80 PS25MSTID\_L Register (Offset = 3C8h) [Reset = 00000000h]

PS25MSTID\_L is shown in [Table 3-518](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register25\_L

**Table 3-518. PS25MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.81 PS25MSTID\_H Register (Offset = 3CCh) [Reset = 00000000h]

PS25MSTID\_H is shown in [Table 3-519](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register25\_H

**Table 3-519. PS25MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.82 PS26MSTID\_L Register (Offset = 3D0h) [Reset = 00000000h]

PS26MSTID\_L is shown in [Table 3-520](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register26\_L

**Table 3-520. PS26MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.83 PS26MSTID\_H Register (Offset = 3D4h) [Reset = 00000000h]

PS26MSTID\_H is shown in [Table 3-521](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register26\_H

**Table 3-521. PS26MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.84 PS27MSTID\_L Register (Offset = 3D8h) [Reset = 00000000h]

PS27MSTID\_L is shown in [Table 3-522](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register27\_L

**Table 3-522. PS27MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.85 PS27MSTID\_H Register (Offset = 3DCh) [Reset = 00000000h]

PS27MSTID\_H is shown in [Table 3-523](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register27\_H

**Table 3-523. PS27MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.86 PS28MSTID\_L Register (Offset = 3E0h) [Reset = 00000000h]

PS28MSTID\_L is shown in [Table 3-524](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register28\_L

**Table 3-524. PS28MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.87 PS28MSTID\_H Register (Offset = 3E4h) [Reset = 00000000h]

PS28MSTID\_H is shown in [Table 3-525](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register28\_H

**Table 3-525. PS28MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.88 PS29MSTID\_L Register (Offset = 3E8h) [Reset = 00000000h]

PS29MSTID\_L is shown in [Table 3-526](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register29\_L

**Table 3-526. PS29MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.89 PS29MSTID\_H Register (Offset = 3ECh) [Reset = 00000000h]

PS29MSTID\_H is shown in [Table 3-527](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register29\_H

**Table 3-527. PS29MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.90 PS30MSTID\_L Register (Offset = 3F0h) [Reset = 00000000h]

PS30MSTID\_L is shown in [Table 3-528](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register30\_L

**Table 3-528. PS30MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.91 PS30MSTID\_H Register (Offset = 3F4h) [Reset = 00000000h]

PS30MSTID\_H is shown in [Table 3-529](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register30\_H

**Table 3-529. PS30MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.92 PS31MSTID\_L Register (Offset = 3F8h) [Reset = 00000000h]

PS31MSTID\_L is shown in [Table 3-530](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register31\_L

**Table 3-530. PS31MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.93 PS31MSTID\_H Register (Offset = 3FCh) [Reset = 00000000h]

PS31MSTID\_H is shown in [Table 3-531](#).

Return to the [Summary Table](#).

Peripheral Frame Master-ID Protection Register31\_H

**Table 3-531. PS31MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PS31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PS31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PS frame. These bits sets the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, peripheral mapped in Quad0 can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15 (b) If bits 31:16 is 1100_1100_1100_1100, peripheral mapped in Quad1 can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15 Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by masters with matching Master-ID. 0 = The peripheral is locked for masters with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.94 PPS0MSTID\_L Register (Offset = 400h) [Reset = 00000000h]

PPS0MSTID\_L is shown in [Table 3-532](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register0\_L

**Table 3-532. PPS0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.95 PPS0MSTID\_H Register (Offset = 404h) [Reset = 00000000h]

PPS0MSTID\_H is shown in [Table 3-533](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register0\_H

**Table 3-533. PPS0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.96 PPS1MSTID\_L Register (Offset = 408h) [Reset = 00000000h]

PPS1MSTID\_L is shown in [Table 3-534](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register1\_L

**Table 3-534. PPS1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.97 PPS1MSTID\_H Register (Offset = 40Ch) [Reset = 00000000h]

PPS1MSTID\_H is shown in [Table 3-535](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register1\_H

**Table 3-535. PPS1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.98 PPS2MSTID\_L Register (Offset = 410h) [Reset = 00000000h]

PPS2MSTID\_L is shown in [Table 3-536](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register2\_L

**Table 3-536. PPS2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.99 PPS2MSTID\_H Register (Offset = 414h) [Reset = 00000000h]

PPS2MSTID\_H is shown in [Table 3-537](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register2\_H

**Table 3-537. PPS2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.100 PPS3MSTID\_L Register (Offset = 418h) [Reset = 00000000h]

PPS3MSTID\_L is shown in [Table 3-538](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register3\_L

**Table 3-538. PPS3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.101 PPS3MSTID\_H Register (Offset = 41Ch) [Reset = 00000000h]

PPS3MSTID\_H is shown in [Table 3-539](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register3\_H

**Table 3-539. PPS3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.102 PPS4MSTID\_L Register (Offset = 420h) [Reset = 00000000h]

PPS4MSTID\_L is shown in [Table 3-540](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register4\_L

**Table 3-540. PPS4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.103 PPS4MSTID\_H Register (Offset = 424h) [Reset = 00000000h]

PPS4MSTID\_H is shown in [Table 3-541](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register4\_H

**Table 3-541. PPS4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.104 PPS5MSTID\_L Register (Offset = 428h) [Reset = 00000000h]

PPS5MSTID\_L is shown in [Table 3-542](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register5\_L

**Table 3-542. PPS5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.105 PPS5MSTID\_H Register (Offset = 42Ch) [Reset = 00000000h]

PPS5MSTID\_H is shown in [Table 3-543](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register5\_H

**Table 3-543. PPS5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.106 PPS6MSTID\_L Register (Offset = 430h) [Reset = 00000000h]

PPS6MSTID\_L is shown in [Table 3-544](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register6\_L

**Table 3-544. PPS6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.107 PPS6MSTID\_H Register (Offset = 434h) [Reset = 00000000h]

PPS6MSTID\_H is shown in [Table 3-545](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register6\_H

**Table 3-545. PPS6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.108 PPS7MSTID\_L Register (Offset = 438h) [Reset = 00000000h]

PPS7MSTID\_L is shown in [Table 3-546](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register7\_L

**Table 3-546. PPS7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.109 PPS7MSTID\_H Register (Offset = 43Ch) [Reset = 00000000h]

PPS7MSTID\_H is shown in [Table 3-547](#).

Return to the [Summary Table](#).

Privileged Peripheral Frame Master-ID Protection Register7\_H

**Table 3-547. PPS7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPS7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPS7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPS frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID register in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.110 PPSE0MSTID\_L Register (Offset = 440h) [Reset = 00000000h]

PPSE0MSTID\_L is shown in [Table 3-548](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register0\_L

**Table 3-548. PPSE0MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE0_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE0_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.111 PPSE0MSTID\_H Register (Offset = 444h) [Reset = 00000000h]

PPSE0MSTID\_H is shown in [Table 3-549](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register0\_H

**Table 3-549. PPSE0MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE0_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE0_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.112 PPSE1MSTID\_L Register (Offset = 448h) [Reset = 00000000h]

PPSE1MSTID\_L is shown in [Table 3-550](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register1\_L

**Table 3-550. PPSE1MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE1_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE1_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.113 PPSE1MSTID\_H Register (Offset = 44Ch) [Reset = 00000000h]

PPSE1MSTID\_H is shown in [Table 3-551](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register1\_H

**Table 3-551. PPSE1MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE1_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE1_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.114 PPSE2MSTID\_L Register (Offset = 450h) [Reset = 00000000h]

PPSE2MSTID\_L is shown in [Table 3-552](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register2\_L

**Table 3-552. PPSE2MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE2_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE2_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.115 PPSE2MSTID\_H Register (Offset = 454h) [Reset = 00000000h]

PPSE2MSTID\_H is shown in [Table 3-553](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register2\_H

**Table 3-553. PPSE2MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE2_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE2_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.116 PPSE3MSTID\_L Register (Offset = 458h) [Reset = 00000000h]

PPSE3MSTID\_L is shown in [Table 3-554](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register3\_L

**Table 3-554. PPSE3MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE3_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE3_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.117 PPSE3MSTID\_H Register (Offset = 45Ch) [Reset = 00000000h]

PPSE3MSTID\_H is shown in [Table 3-555](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register3\_H

**Table 3-555. PPSE3MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE3_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE3_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.118 PPSE4MSTID\_L Register (Offset = 460h) [Reset = 00000000h]

PPSE4MSTID\_L is shown in [Table 3-556](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register4\_L

**Table 3-556. PPSE4MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE4_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE4_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.119 PPSE4MSTID\_H Register (Offset = 464h) [Reset = 00000000h]

PPSE4MSTID\_H is shown in [Table 3-557](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register4\_H

**Table 3-557. PPSE4MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE4_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE4_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.120 PPSE5MSTID\_L Register (Offset = 468h) [Reset = 00000000h]

PPSE5MSTID\_L is shown in [Table 3-558](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register5\_L

**Table 3-558. PPSE5MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE5_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE5_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.121 PPSE5MSTID\_H Register (Offset = 46Ch) [Reset = 00000000h]

PPSE5MSTID\_H is shown in [Table 3-559](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register5\_H

**Table 3-559. PPSE5MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE5_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE5_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.122 PPSE6MSTID\_L Register (Offset = 470h) [Reset = 00000000h]

PPSE6MSTID\_L is shown in [Table 3-560](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register6\_L

**Table 3-560. PPSE6MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE6_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE6_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.123 PPSE6MSTID\_H Register (Offset = 474h) [Reset = 00000000h]

PPSE6MSTID\_H is shown in [Table 3-561](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register6\_H

**Table 3-561. PPSE6MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE6_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE6_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.124 PPSE7MSTID\_L Register (Offset = 478h) [Reset = 00000000h]

PPSE7MSTID\_L is shown in [Table 3-562](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register7\_L

**Table 3-562. PPSE7MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE7_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE7_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.125 PPSE7MSTID\_H Register (Offset = 47Ch) [Reset = 00000000h]

PPSE7MSTID\_H is shown in [Table 3-563](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register7\_H

**Table 3-563. PPSE7MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE7_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE7_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.126 PPSE8MSTID\_L Register (Offset = 480h) [Reset = 00000000h]

PPSE8MSTID\_L is shown in [Table 3-564](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register8\_L

**Table 3-564. PPSE8MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE8_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE8_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.127 PPSE8MSTID\_H Register (Offset = 484h) [Reset = 00000000h]

PPSE8MSTID\_H is shown in [Table 3-565](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register8\_H

**Table 3-565. PPSE8MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE8_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE8_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.128 PPSE9MSTID\_L Register (Offset = 488h) [Reset = 00000000h]

PPSE9MSTID\_L is shown in [Table 3-566](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register9\_L

**Table 3-566. PPSE9MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE9_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE9_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.129 PPSE9MSTID\_H Register (Offset = 48Ch) [Reset = 00000000h]

PPSE9MSTID\_H is shown in [Table 3-567](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register9\_H

**Table 3-567. PPSE9MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE9_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE9_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.130 PPSE10MSTID\_L Register (Offset = 490h) [Reset = 00000000h]

PPSE10MSTID\_L is shown in [Table 3-568](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register10\_L

**Table 3-568. PPSE10MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE10_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE10_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.131 PPSE10MSTID\_H Register (Offset = 494h) [Reset = 00000000h]

PPSE10MSTID\_H is shown in [Table 3-569](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register10\_H

**Table 3-569. PPSE10MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE10_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE10_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.132 PPSE11MSTID\_L Register (Offset = 498h) [Reset = 00000000h]

PPSE11MSTID\_L is shown in [Table 3-570](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register11\_L

**Table 3-570. PPSE11MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE11_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE11_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.133 PPSE11MSTID\_H Register (Offset = 49Ch) [Reset = 00000000h]

PPSE11MSTID\_H is shown in [Table 3-571](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register11\_H

**Table 3-571. PPSE11MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE11_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE11_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.134 PPSE12MSTID\_L Register (Offset = 4A0h) [Reset = 00000000h]

PPSE12MSTID\_L is shown in [Table 3-572](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register12\_L

**Table 3-572. PPSE12MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE12_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE12_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.135 PPSE12MSTID\_H Register (Offset = 4A4h) [Reset = 00000000h]

PPSE12MSTID\_H is shown in [Table 3-573](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register12\_H

**Table 3-573. PPSE12MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE12_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE12_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.136 PPSE13MSTID\_L Register (Offset = 4A8h) [Reset = 00000000h]

PPSE13MSTID\_L is shown in [Table 3-574](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register13\_L

**Table 3-574. PPSE13MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE13_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE13_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.137 PPSE13MSTID\_H Register (Offset = 4ACh) [Reset = 00000000h]

PPSE13MSTID\_H is shown in [Table 3-575](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register13\_H

**Table 3-575. PPSE13MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE13_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE13_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.138 PPSE14MSTID\_L Register (Offset = 4B0h) [Reset = 00000000h]

PPSE14MSTID\_L is shown in [Table 3-576](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register14\_L

**Table 3-576. PPSE14MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE14_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE14_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.139 PPSE14MSTID\_H Register (Offset = 4B4h) [Reset = 00000000h]

PPSE14MSTID\_H is shown in [Table 3-577](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register14\_H

**Table 3-577. PPSE14MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE14_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE14_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.140 PPSE15MSTID\_L Register (Offset = 4B8h) [Reset = 00000000h]

PPSE15MSTID\_L is shown in [Table 3-578](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register15\_L

**Table 3-578. PPSE15MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE15_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE15_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.141 PPSE15MSTID\_H Register (Offset = 4BCh) [Reset = 00000000h]

PPSE15MSTID\_H is shown in [Table 3-579](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register15\_H

**Table 3-579. PPSE15MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE15_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE15_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.142 PPSE16MSTID\_L Register (Offset = 4C0h) [Reset = 00000000h]

PPSE16MSTID\_L is shown in [Table 3-580](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register16\_L

**Table 3-580. PPSE16MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE16_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE16_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.143 PPSE16MSTID\_H Register (Offset = 4C4h) [Reset = 00000000h]

PPSE16MSTID\_H is shown in [Table 3-581](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register16\_H

**Table 3-581. PPSE16MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE16_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE16_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.144 PPSE17MSTID\_L Register (Offset = 4C8h) [Reset = 00000000h]

PPSE17MSTID\_L is shown in [Table 3-582](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register17\_L

**Table 3-582. PPSE17MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE17_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE17_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.145 PPSE17MSTID\_H Register (Offset = 4CCh) [Reset = 00000000h]

PPSE17MSTID\_H is shown in [Table 3-583](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register17\_H

**Table 3-583. PPSE17MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE17_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE17_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.146 PPSE18MSTID\_L Register (Offset = 4D0h) [Reset = 00000000h]

PPSE18MSTID\_L is shown in [Table 3-584](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register18\_L

**Table 3-584. PPSE18MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE18_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE18_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.147 PPSE18MSTID\_H Register (Offset = 4D4h) [Reset = 00000000h]

PPSE18MSTID\_H is shown in [Table 3-585](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register18\_H

**Table 3-585. PPSE18MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE18_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE18_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.148 PPSE19MSTID\_L Register (Offset = 4D8h) [Reset = 00000000h]

PPSE19MSTID\_L is shown in [Table 3-586](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register19\_L

**Table 3-586. PPSE19MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE19_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE19_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.149 PPSE19MSTID\_H Register (Offset = 4DCh) [Reset = 00000000h]

PPSE19MSTID\_H is shown in [Table 3-587](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register19\_H

**Table 3-587. PPSE19MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE19_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE19_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.150 PPSE20MSTID\_L Register (Offset = 4E0h) [Reset = 00000000h]

PPSE20MSTID\_L is shown in [Table 3-588](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register20\_L

**Table 3-588. PPSE20MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE20_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE20_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.151 PPSE20MSTID\_H Register (Offset = 4E4h) [Reset = 00000000h]

PPSE20MSTID\_H is shown in [Table 3-589](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register20\_H

**Table 3-589. PPSE20MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE20_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE20_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.152 PPSE21MSTID\_L Register (Offset = 4E8h) [Reset = 00000000h]

PPSE21MSTID\_L is shown in [Table 3-590](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register21\_L

**Table 3-590. PPSE21MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE21_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE21_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.153 PPSE21MSTID\_H Register (Offset = 4ECh) [Reset = 00000000h]

PPSE21MSTID\_H is shown in [Table 3-591](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register21\_H

**Table 3-591. PPSE21MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE21_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE21_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.154 PPSE22MSTID\_L Register (Offset = 4F0h) [Reset = 00000000h]

PPSE22MSTID\_L is shown in [Table 3-592](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register22\_L

**Table 3-592. PPSE22MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE22_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE22_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.155 PPSE22MSTID\_H Register (Offset = 4F4h) [Reset = 00000000h]

PPSE22MSTID\_H is shown in [Table 3-593](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register22\_H

**Table 3-593. PPSE22MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE22_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE22_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.156 PPSE23MSTID\_L Register (Offset = 4F8h) [Reset = 00000000h]

PPSE23MSTID\_L is shown in [Table 3-594](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register23\_L

**Table 3-594. PPSE23MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE23_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE23_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.157 PPSE23MSTID\_H Register (Offset = 4FCh) [Reset = 00000000h]

PPSE23MSTID\_H is shown in [Table 3-595](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register23\_H

**Table 3-595. PPSE23MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE23_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE23_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.158 PPSE24MSTID\_L Register (Offset = 500h) [Reset = 00000000h]

PPSE24MSTID\_L is shown in [Table 3-596](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register24\_L

**Table 3-596. PPSE24MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE24_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE24_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.159 PPSE24MSTID\_H Register (Offset = 504h) [Reset = 00000000h]

PPSE24MSTID\_H is shown in [Table 3-597](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register24\_H

**Table 3-597. PPSE24MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE24_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE24_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.160 PPSE25MSTID\_L Register (Offset = 508h) [Reset = 00000000h]

PPSE25MSTID\_L is shown in [Table 3-598](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register25\_L

**Table 3-598. PPSE25MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE25_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE25_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.161 PPSE25MSTID\_H Register (Offset = 50Ch) [Reset = 00000000h]

PPSE25MSTID\_H is shown in [Table 3-599](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register25\_H

**Table 3-599. PPSE25MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE25_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE25_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.162 PPSE26MSTID\_L Register (Offset = 510h) [Reset = 00000000h]

PPSE26MSTID\_L is shown in [Table 3-600](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register26\_L

**Table 3-600. PPSE26MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE26_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE26_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.163 PPSE26MSTID\_H Register (Offset = 514h) [Reset = 00000000h]

PPSE26MSTID\_H is shown in [Table 3-601](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register26\_H

**Table 3-601. PPSE26MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE26_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE26_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.164 PPSE27MSTID\_L Register (Offset = 518h) [Reset = 00000000h]

PPSE27MSTID\_L is shown in [Table 3-602](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register27\_L

**Table 3-602. PPSE27MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE27_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE27_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.165 PPSE27MSTID\_H Register (Offset = 51Ch) [Reset = 00000000h]

PPSE27MSTID\_H is shown in [Table 3-603](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register27\_H

**Table 3-603. PPSE27MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE27_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE27_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.166 PPSE28MSTID\_L Register (Offset = 520h) [Reset = 00000000h]

PPSE28MSTID\_L is shown in [Table 3-604](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register28\_L

**Table 3-604. PPSE28MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE28_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE28_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.167 PPSE28MSTID\_H Register (Offset = 524h) [Reset = 00000000h]

PPSE28MSTID\_H is shown in [Table 3-605](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register28\_H

**Table 3-605. PPSE28MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE28_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE28_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.168 PPSE29MSTID\_L Register (Offset = 528h) [Reset = 00000000h]

PPSE29MSTID\_L is shown in [Table 3-606](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register29\_L

**Table 3-606. PPSE29MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE29_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE29_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.169 PPSE29MSTID\_H Register (Offset = 52Ch) [Reset = 00000000h]

PPSE29MSTID\_H is shown in [Table 3-607](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register29\_H

**Table 3-607. PPSE29MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE29_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE29_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.170 PPSE30MSTID\_L Register (Offset = 530h) [Reset = 00000000h]

PPSE30MSTID\_L is shown in [Table 3-608](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register30\_L

**Table 3-608. PPSE30MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE30_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE30_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.171 PPSE30MSTID\_H Register (Offset = 534h) [Reset = 00000000h]

PPSE30MSTID\_H is shown in [Table 3-609](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register30\_H

**Table 3-609. PPSE30MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE30_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE30_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.172 PPSE31MSTID\_L Register (Offset = 538h) [Reset = 00000000h]

PPSE31MSTID\_L is shown in [Table 3-610](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register31\_L

**Table 3-610. PPSE31MSTID\_L Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE31_QUAD1_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE31_QUAD0_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.173 PPSE31MSTID\_H Register (Offset = 53Ch) [Reset = 00000000h]

PPSE31MSTID\_H is shown in [Table 3-611](#).

Return to the [Summary Table](#).

Privileged Peripheral Extended Frame Master-ID Protection Register31\_H

**Table 3-611. PPSE31MSTID\_H Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPSE31_QUAD3_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPSE31_QUAD2_MSTID	R/W	0h	There are 16 bits for each quadrant in PPSE frame. These bits set the permission for maximum of 16 masters to address the peripheral mapped in each of the quadrant. The scheme is similar to the one described for PS MSTID in section 1.7.30. Readable in both user and privileged modes. 1 = The peripheral mapped in the quadrant can be addressed by master with matching Master-ID. 0 = The peripheral is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.174 PCS0MSTID Register (Offset = 540h) [Reset = 00000000h]

PCS0MSTID is shown in [Table 3-612](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register0

**Table 3-612. PCS0MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS1MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS0MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.175 PCS1MSTID Register (Offset = 544h) [Reset = 00000000h]

PCS1MSTID is shown in [Table 3-613](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register1

**Table 3-613. PCS1MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS3MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS2MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.176 PCS2MSTID Register (Offset = 548h) [Reset = 00000000h]

PCS2MSTID is shown in [Table 3-614](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register2

**Table 3-614. PCS2MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS5MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS4MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.177 PCS3MSTID Register (Offset = 54Ch) [Reset = 00000000h]

PCS3MSTID is shown in [Table 3-615](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register3

**Table 3-615. PCS3MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS7MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS6MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.178 PCS4MSTID Register (Offset = 550h) [Reset = 00000000h]

PCS4MSTID is shown in [Table 3-616](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register4

**Table 3-616. PCS4MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS9MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS8MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.3.179 PCS5MSTID Register (Offset = 554h) [Reset = 00000000h]

PCS5MSTID is shown in [Table 3-617](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register5

**Table 3-617. PCS5MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS11MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS10MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.180 PCS6MSTID Register (Offset = 558h) [Reset = 00000000h]

PCS6MSTID is shown in [Table 3-618](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register6

**Table 3-618. PCS6MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS13MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS12MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.181 PCS7MSTID Register (Offset = 55Ch) [Reset = 00000000h]

PCS7MSTID is shown in [Table 3-619](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register7

**Table 3-619. PCS7MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS15MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS14MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.182 PCS8MSTID Register (Offset = 560h) [Reset = 00000000h]

PCS8MSTID is shown in [Table 3-620](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register8

**Table 3-620. PCS8MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS17MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS16MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.183 PCS9MSTID Register (Offset = 564h) [Reset = 00000000h]

PCS9MSTID is shown in [Table 3-621](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register9

**Table 3-621. PCS9MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS19MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS18MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.184 PCS10MSTID Register (Offset = 568h) [Reset = 00000000h]

PCS10MSTID is shown in [Table 3-622](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register10

**Table 3-622. PCS10MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS21MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS20MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.185 PCS11MSTID Register (Offset = 56Ch) [Reset = 00000000h]

PCS11MSTID is shown in [Table 3-623](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register11

**Table 3-623. PCS11MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS23MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS22MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.186 PCS12MSTID Register (Offset = 570h) [Reset = 00000000h]

PCS12MSTID is shown in [Table 3-624](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register12

**Table 3-624. PCS12MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS25MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS24MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.3.187 PCS13MSTID Register (Offset = 574h) [Reset = 00000000h]

PCS13MSTID is shown in [Table 3-625](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register13

**Table 3-625. PCS13MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS27MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS26MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.188 PCS14MSTID Register (Offset = 578h) [Reset = 00000000h]

PCS14MSTID is shown in [Table 3-626](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register14

**Table 3-626. PCS14MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS29MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS28MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.189 PCS15MSTID Register (Offset = 57Ch) [Reset = 00000000h]

PCS15MSTID is shown in [Table 3-627](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register15

**Table 3-627. PCS15MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS31MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS30MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.190 PCS16MSTID Register (Offset = 580h) [Reset = 00000000h]

PCS16MSTID is shown in [Table 3-628](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register16

**Table 3-628. PCS16MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS33MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS32MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.191 PCS17MSTID Register (Offset = 584h) [Reset = 00000000h]

PCS17MSTID is shown in [Table 3-629](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register17

**Table 3-629. PCS17MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS35MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS34MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.192 PCS18MSTID Register (Offset = 588h) [Reset = 00000000h]

PCS18MSTID is shown in [Table 3-630](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register18

**Table 3-630. PCS18MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS37MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS36MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.193 PCS19MSTID Register (Offset = 58Ch) [Reset = 00000000h]

PCS19MSTID is shown in [Table 3-631](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register19

**Table 3-631. PCS19MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS39MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS38MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.194 PCS20MSTID Register (Offset = 590h) [Reset = 00000000h]

PCS20MSTID is shown in [Table 3-632](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register20

**Table 3-632. PCS20MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS41MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS40MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.3.195 PCS21MSTID Register (Offset = 594h) [Reset = 00000000h]

PCS21MSTID is shown in [Table 3-633](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register21

**Table 3-633. PCS21MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS43MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS42MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.196 PCS22MSTID Register (Offset = 598h) [Reset = 00000000h]

PCS22MSTID is shown in [Table 3-634](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register22

**Table 3-634. PCS22MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS45MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS44MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.197 PCS23MSTID Register (Offset = 59Ch) [Reset = 00000000h]

PCS23MSTID is shown in [Table 3-635](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register23

**Table 3-635. PCS23MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS47MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS46MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.198 PCS24MSTID Register (Offset = 5A0h) [Reset = 00000000h]

PCS24MSTID is shown in [Table 3-636](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register24

**Table 3-636. PCS24MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS49MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS48MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.199 PCS25MSTID Register (Offset = 5A4h) [Reset = 00000000h]

PCS25MSTID is shown in [Table 3-637](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register25

**Table 3-637. PCS25MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS51MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS50MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.200 PCS26MSTID Register (Offset = 5A8h) [Reset = 00000000h]

PCS26MSTID is shown in [Table 3-638](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register26

**Table 3-638. PCS26MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS53MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS52MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.201 PCS27MSTID Register (Offset = 5ACh) [Reset = 00000000h]

PCS27MSTID is shown in [Table 3-639](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register27

**Table 3-639. PCS27MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS55MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS54MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.202 PCS28MSTID Register (Offset = 5B0h) [Reset = 00000000h]

PCS28MSTID is shown in [Table 3-640](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register28

**Table 3-640. PCS28MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS57MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS56MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0



### 3.4.3.203 PCS29MSTID Register (Offset = 5B4h) [Reset = 00000000h]

PCS29MSTID is shown in [Table 3-641](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register29

**Table 3-641. PCS29MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS59MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS58MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.204 PCS30MSTID Register (Offset = 5B8h) [Reset = 00000000h]

PCS30MSTID is shown in [Table 3-642](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register30

**Table 3-642. PCS30MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS61MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS60MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.205 PCS31MSTID Register (Offset = 5BCh) [Reset = 00000000h]

PCS31MSTID is shown in [Table 3-643](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register31

**Table 3-643. PCS31MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PCS63MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0
15-0	PCS62MSTID	R/W	0h	There are 16 bits for each frame in PCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The following examples shows the usage of these register bits. (a) If bits 15:0 is 1010_1010_1010_1010, memory frame mapped to PCSm can be addressed by Masters with Master-ID equals 1,3,5,7,9,11,13,15. (b) If bits 31:24 is 1100_1100_1100_1100, memory frame mapped to PCS(m+1) can be addressed by Masters with Master-ID equals 2,3,6,7,10,11,14,15. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERRORr. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0

### 3.4.3.206 PPCS0MSTID Register (Offset = 5C0h) [Reset = 00000000h]

PPCS0MSTID is shown in [Table 3-644](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register32

**Table 3-644. PPCS0MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS1MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS0MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.207 PPCS1MSTID Register (Offset = 5C4h) [Reset = 00000000h]

PPCS1MSTID is shown in [Table 3-645](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register33

**Table 3-645. PPCS1MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS3MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS2MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.208 PPCS2MSTID Register (Offset = 5C8h) [Reset = 00000000h]

PPCS2MSTID is shown in [Table 3-646](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register34

**Table 3-646. PPCS2MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS5MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS4MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.209 PPCS3MSTID Register (Offset = 5CCh) [Reset = 00000000h]

PPCS3MSTID is shown in [Table 3-647](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register35

**Table 3-647. PPCS3MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS7MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS6MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.210 PPCS4MSTID Register (Offset = 5D0h) [Reset = 00000000h]

PPCS4MSTID is shown in [Table 3-648](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register36

**Table 3-648. PPCS4MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS9MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS8MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



### 3.4.3.211 PPCS5MSTID Register (Offset = 5D4h) [Reset = 00000000h]

PPCS5MSTID is shown in [Table 3-649](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register37

**Table 3-649. PPCS5MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS11MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS10MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.212 PPCS6MSTID Register (Offset = 5D8h) [Reset = 00000000h]

PPCS6MSTID is shown in [Table 3-650](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register38

**Table 3-650. PPCS6MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS13MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS12MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.213 PPCS7MSTID Register (Offset = 5DCh) [Reset = 00000000h]

PPCS7MSTID is shown in [Table 3-651](#).

Return to the [Summary Table](#).

Memory Frame Master ID Protection Register39

**Table 3-651. PPCS7MSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PPCS15MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.
15-0	PPCS14MSTID	R/W	0h	There are 16 bits for each frame in PPCS. These bits sets the permission for maximum of 16 masters to address the memory mapped in each of the frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.

### 3.4.3.214 PCREXTMSTID Register (Offset = 5E0h) [Reset = 00000000h]

PCREXTMSTID is shown in [Table 3-652](#).

Return to the [Summary Table](#).

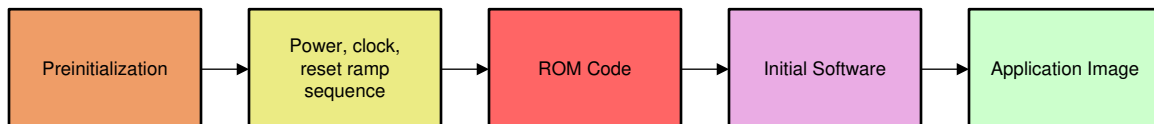
Master-ID Protection Register for external PCR

**Table 3-652. PCREXTMSTID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PCREXT_MSTID	R/W	0h	These bits sets the permission for maximum of 16 masters to address the external PCR frame. The scheme is similar to the one described for PCSm MSTID in section 1.7.33. Readable in both user and privileged modes. 1 = The memory mapped in respective frames can be addressed by master with matching Master-ID. 0 = The memory is locked for master with matching Master-ID. PCR responds with AERROR. Writable only in privileged mode 1 = Sets the corresponding bit. 0 = Clears the corresponding bit. Writes to unimplemented bits have no effect and reads yield 0.



## 4.1



**Figure 4-1. Device Initialization**

The first two steps in the initialization process are hardware-oriented; however, they require an understanding of the process of configuring these system interface pins (balls on the device), which have software-configurable functionality. This configuration is an essential part of the chip configuration and is application-dependent. This chapter discusses these system-interface pins, the associated configuration registers, and memory structures that are vital to the correct initialization of the device.

## 4.2 Boot Process

### 4.2.1 ROM Code Overview

To accommodate various system scenarios, the ROM code supports several boot modes. These boot modes can be broadly classified as:

- Host boot modes
- Memory boot modes.

During a host boot, the device is configured to receive code from a host through the selected interface (such as UART). Either the host writes the application code directly into internal memory over UART, or the ROM code receives the application code on the selected interface and stores it in internal memory.

During a memory boot, the device transfers code from non-volatile memory to internal memory for execution.

In all boot modes, the entire boot operation can be partitioned into two sections:

- Hardware initialization phase
- Boot process

During initialization, the ROM code configures the device resources (PLLs, peripherals, pins) as needed to support the boot process. The resources used depend on the boot mode requirements. During the boot process, the boot image can be loaded into device memory and executed, depending on the boot peripheral. HSM ROM code performs code verification and allows or forbids the image execution.

The main configuration source for boot after power-up are the SOP mode pins sampled automatically after reset release and stored in device status registers. At ROM code startup, these pin values are read from the registers to create the boot peripheral list, and the boot configuration tables used later to initialize and startup the PLLs and boot peripherals.

### 4.2.2 Boot Modes

MCU ROM supports the functionality of loading the secondary boot loader (SBL). The SBL can be loaded through QSPI (primary/secondary SBL) and UART mode. The primary use case of QSPI programming through UART interface is to support the uniFlash utility. This mode programming may be applicable for certain use

cases, such as initial FLASH programming in volume mass production, that can also be achieved with special in-circuit gang programming tools.

Primary functional boot mode is through QSPI FLASH. MCU ROM supports managing multiple (primary and backup) QSPI SBL images. It can identify the primary image, and switch to secondary image load if primary image load fails.

**Table 4-1. Boot Modes and Boot Media**

Boot Mode/Peripheral	Boot Media/Host	Notes
QSPI	QSPI flash	Download and boot SBL from QSPI flash. Attempt Primary SBL, followed by Secondary SBL if primary loading fails: If above is not successful
UART	External host	Download and boot SBL from UART. Device is expected to get SBL from UART. We will support the XMODEM protocol for download over UART.

For more information about various bootmode and flash device support please refer to [application note](#).

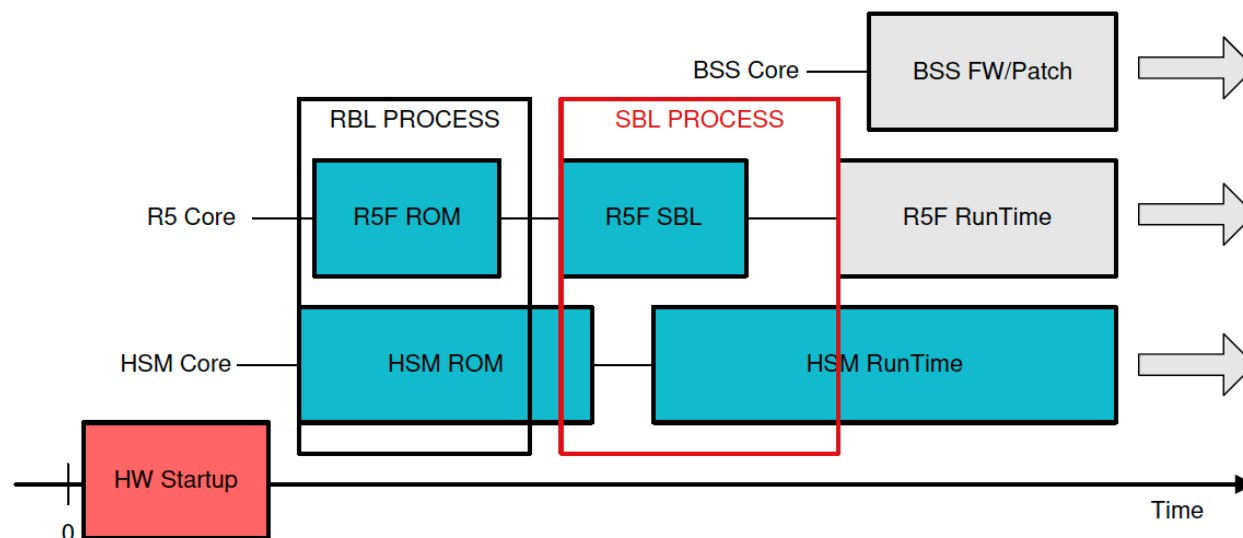
### 4.2.3 SOP Mode Pins

[Table 4-2](#) lists the functional mode pin settings to be done for the SOP lines of the device to boot using different peripheral. New SOP modes will be added later.

**Table 4-2. Functional Mode Pin Settings**

Functional Mode	SOP Mode	PMIC_OUT	SYNC_OUT	TDO
Func Mode –QSPI	SOP_MODE4	0	0	1
Func Mode –UART	SOP_MODE5	1	0	1

### 4.2.4 BOOT-ROM Architecture (RBL)


**Figure 4-3. R5F SBL**
**Figure 4-2. Boot Flow and Boot ROM Architecture**

The RBL process goal is to load, verify, optionally decrypt, and launch an authentic R5F software image that accomplishes general-purpose/secure boot goals. The RBL process is implemented jointly by the R5F and HSM ROM as illustrated in [Figure 4-2](#).

## HSM ROM

This HSM ROM contains the first set of software instructions that is executed by any processor core on all AWR2544 devices. All state changes in the device after the external reset is released but prior to the beginning of the HSM ROM code execution are purely a function of hardware logic. This hardware logic must perform sufficient ramp up and initialization to allow the Cortex M4 core of the HSM to leave reset and begin execution from its reset vector. The HSM ROM code is “time zero” software. The HSM ROM code is only intended to be used during the initial load of the secondary boot loader (SBL) and HSM RunTime image. HSM ROM is also responsible for providing Test/Debug capabilities when functional boot is interrupted.

## R5F ROM

The R5F ROM code is only intended to be used during the initial load of the secondary boot loader (SBL). RBL can load the SBL content only to MSS\_L2 area and at the end copies SBL's IVT (interrupt vector table) of 640B size to TCMA\_RAM\_CR5A before switching to SBL. This ROM-to-RAM (RBL to SBL) switching causes TCMA\_ROM\_CR5A eclipse to TCMA\_RAM\_CR5A i.e. 0x0000\_0000 now maps to TCMA\_RAM\_CR5A memory area. Use of RBL at any other time in the lifecycle of a system is not supported. R5F ROM executes set of self-tests: PBIST (DATA SRAM, PROGRAM SRAM, and ROM Code Integrity) using a hardware-defined interface. If unsuccessful, the error condition is indicated, external ESM signal asserted, and boot does not proceed.

## R5F SBL

This SBL can perform complete boot sequence on general purpose devices. Customers are expected to develop their own SBLs, supporting a wider range of requirements (such as different interfaces, additional protocols, different image formats, future update flow and so forth).

## HSM RunTime/R5F Runtime

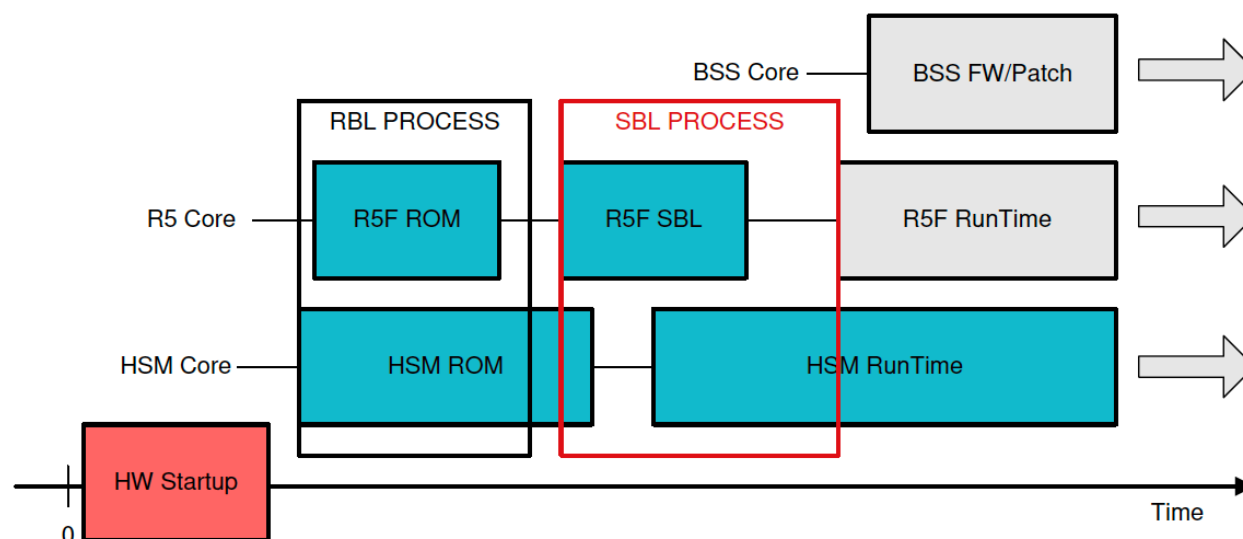
HSM RunTime is only applicable in case of Secured (HS) variant and out of scope of this document. R5F user developed applications and loaded/executed by SBL.

### 4.2.5 R5F SBL Loading

The SBL process starts with R5F executing loaded and verified R5F SBL code.

The only service exposed to R5F SBL by HSM ROM is the API to load the HSM RunTime. All other HSM services are outside of the HSM ROM. The R5 Boot ROM and HSM Boot ROM work together to load the R5F SBL. When the R5F SBL has been successfully validated and loaded, the R5F Boot-ROM is eclipsed and the R5F SBL executes.

[Figure 4-3](#) is the simplified sequence of the boot process. It is provided for illustration purposes only. An illustration of the SBL and HSM RunTime is similar to what is shown in [Figure 4-3](#).



**Figure 4-3. R5F SBL**

**Figure 4-3. R5F SBL**

HSM ROM configures the IPC interface, unhalts the R5F core, and then waits for a confirmation message from the R5F Core (over IPC). The R5F core sends a confirmation to the HSM ROM. The R5F ROM then configures the QSPI interface and reads the R5F SBL image from FLASH. The SBL image might or might not have integrity check enabled (based on the certificate). HSM ROM switches the memory map for R5F from ROM to RAM and resets the R5F to execute the SBL image. The HSM-ROM for GP devices is now locked (black-boxed). SBL can now load the multicore image (containing MSS & DSS application and RSS firmware/patch images) to the corresponding RAMs and core soft-reset to jump to R5F RunTime. SBL or R5F RunTime can release out-of-reset & unhalt RadarSS and C66x subsystems.

For more information about SBL and bootloading please refer to the [application note](#)

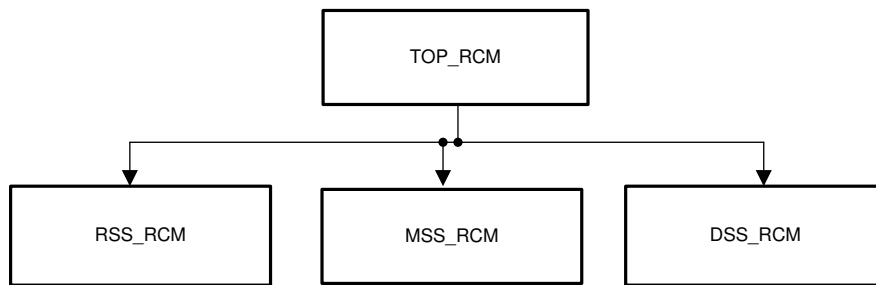




## 5.1 Overview

PRCM manages clocks, resets, and power domain control of subsystems and modules inside the device. Additionally, configuration of certain device-level features is also performed through this module. PRCM has control and status registers to achieve this functionality. The Clock and Reset Management in AWR2544 is distributed. The Main subsystem TOPRCM module controls all the Subsystem Resets and Clocks. The SubSystem RCM modules control their respective subsystem IPs

The available address space of PRCM is divided as in [Figure 5-1](#).



**Figure 5-1. Device Configuration**

**Table 5-1. PRCM Space**

PRCM Space	Description
MSS_TOPRCM/TOP_RCM	Top-level reset, clock management registers
MSS_RCM	Main subsystem reset, clock management registers
RSS_RCM	Radar subsystem reset, clock management registers
DSS_RCM	DSP reset, clock management registers

### Note

TOP\_RCM and MSS\_TOPRCM names are used interchangeably across this document and are pointed to same RCM space.

## 5.2 Control Registers

### 5.2.1 MSS\_TOPRCM Registers

Table 5-2 lists the memory-mapped registers for the MSS\_TOPRCM registers. All register offset addresses not listed in Table 5-2 should be considered as reserved locations and the register contents should not be modified.

**Table 5-2. MSS\_TOPRCM Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.1.1</a>
4h	HW_REG0		<a href="#">Section 5.2.1.2</a>
8h	HW_REG1		<a href="#">Section 5.2.1.3</a>
Ch	PREVIOUS_NAME		<a href="#">Section 5.2.1.4</a>
10h	HW_REG3		<a href="#">Section 5.2.1.5</a>
14h	HSI_CLK_SRC_SEL		<a href="#">Section 5.2.1.6</a>
18h	CSIRX_CLK_SRC_SEL		<a href="#">Section 5.2.1.7</a>
1Ch	MCUCLKOUT_CLK_SRC_SEL		<a href="#">Section 5.2.1.8</a>
20h	PMICCLKOUT_CLK_SRC_SEL		<a href="#">Section 5.2.1.9</a>
24h	OBSCLKOUT_CLK_SRC_SEL		<a href="#">Section 5.2.1.10</a>
28h	TRCCLKOUT_CLK_SRC_SEL		<a href="#">Section 5.2.1.11</a>
44h	CSIRX_DIV_VAL		<a href="#">Section 5.2.1.12</a>
48h	MCUCLKOUT_DIV_VAL		<a href="#">Section 5.2.1.13</a>
4Ch	PMICCLKOUT_DIV_VAL		<a href="#">Section 5.2.1.14</a>
50h	OBSCLKOUT_DIV_VAL		<a href="#">Section 5.2.1.15</a>
54h	TRCCLKOUT_DIV_VAL		<a href="#">Section 5.2.1.16</a>
84h	CSIRX_CLK_GATE		<a href="#">Section 5.2.1.17</a>
88h	MCUCLKOUT_CLK_GATE		<a href="#">Section 5.2.1.18</a>
8Ch	PMICCLKOUT_CLK_GATE		<a href="#">Section 5.2.1.19</a>
90h	OBSCLKOUT_CLK_GATE		<a href="#">Section 5.2.1.20</a>
94h	TRCCLKOUT_CLK_GATE		<a href="#">Section 5.2.1.21</a>
98h	DSS_CLK_GATE		<a href="#">Section 5.2.1.22</a>
C0h	HSI_CLK_STATUS		<a href="#">Section 5.2.1.23</a>
C8h	MCUCLKOUT_CLK_STATUS		<a href="#">Section 5.2.1.24</a>
CCh	PMICCLKOUT_CLK_STATUS		<a href="#">Section 5.2.1.25</a>
D0h	OBSCLKOUT_CLK_STATUS		<a href="#">Section 5.2.1.26</a>
D4h	TRCCLKOUT_CLK_STATUS		<a href="#">Section 5.2.1.27</a>
100h	WARM_RESET_CONFIG		<a href="#">Section 5.2.1.28</a>
104h	SYS_RST_CAUSE		<a href="#">Section 5.2.1.29</a>
108h	SYS_RST_CAUSE_CLR		<a href="#">Section 5.2.1.30</a>
10Ch	DSS_RST_CTRL		<a href="#">Section 5.2.1.31</a>
204h	RS232_BITINTERVAL		<a href="#">Section 5.2.1.32</a>
208h	LVDS_PAD_CTRL0		<a href="#">Section 5.2.1.33</a>
20Ch	LVDS_PAD_CTRL1		<a href="#">Section 5.2.1.34</a>
210h	DFT_DMLED_EXEC		<a href="#">Section 5.2.1.35</a>
214h	DFT_DMLED_STATUS		<a href="#">Section 5.2.1.36</a>
218h	LIMP_MODE_EN		<a href="#">Section 5.2.1.37</a>
21Ch	PMICCLKOUT_DCDC_CTRL		<a href="#">Section 5.2.1.38</a>
220h	PMICCLKOUT_DCDC_SLOPE		<a href="#">Section 5.2.1.39</a>
224h	RCOSC32K_CTRL		<a href="#">Section 5.2.1.40</a>
228h	ANA_HSI2DIGCLK_GATE		<a href="#">Section 5.2.1.41</a>

**Table 5-2. MSS\_TOPRCM Registers (continued)**

Offset	Acronym	Register Name	Section
22Ch	DSI_PHY_PWRCTRL		<a href="#">Section 5.2.1.42</a>
230h	RS232_SOP11_BITINTERVAL		<a href="#">Section 5.2.1.43</a>
400h	PLL_CORE_PWRCTRL		<a href="#">Section 5.2.1.44</a>
404h	PLL_CORE_CLKCTRL		<a href="#">Section 5.2.1.45</a>
408h	PLL_CORE_TENABLE		<a href="#">Section 5.2.1.46</a>
40Ch	PLL_CORE_TENABLEDIV		<a href="#">Section 5.2.1.47</a>
410h	PLL_CORE_M2NDIV		<a href="#">Section 5.2.1.48</a>
414h	PLL_CORE_MN2DIV		<a href="#">Section 5.2.1.49</a>
418h	PLL_CORE_FRACDIV		<a href="#">Section 5.2.1.50</a>
41Ch	PLL_CORE_BWCTRL		<a href="#">Section 5.2.1.51</a>
420h	PLL_CORE_FRACCTRL		<a href="#">Section 5.2.1.52</a>
424h	PLL_CORE_STATUS		<a href="#">Section 5.2.1.53</a>
428h	PLL_CORE_HSDIVIDER		<a href="#">Section 5.2.1.54</a>
42Ch	PLL_CORE_HSDIVIDER_CLKOUT0		<a href="#">Section 5.2.1.55</a>
430h	PLL_CORE_HSDIVIDER_CLKOUT1		<a href="#">Section 5.2.1.56</a>
434h	PLL_CORE_HSDIVIDER_CLKOUT2		<a href="#">Section 5.2.1.57</a>
438h	PLL_CORE_HSDIVIDER_CLKOUT3		<a href="#">Section 5.2.1.58</a>
43Ch	MSS_CR5_CLK_SRC_SEL		<a href="#">Section 5.2.1.59</a>
440h	MSS_CR5_DIV_VAL		<a href="#">Section 5.2.1.60</a>
444h	SYS_CLK_DIV_VAL		<a href="#">Section 5.2.1.61</a>
448h	MSS_CR5_CLK_GATE		<a href="#">Section 5.2.1.62</a>
44Ch	SYS_CLK_GATE		<a href="#">Section 5.2.1.63</a>
450h	SYS_CLK_STATUS		<a href="#">Section 5.2.1.64</a>
454h	MSS_CR5_CLK_STATUS		<a href="#">Section 5.2.1.65</a>
458h	PLL_CORE_RSTCTRL		<a href="#">Section 5.2.1.66</a>
45Ch	PLL_CORE_HSDIVIDER_RSTCTRL		<a href="#">Section 5.2.1.67</a>
460h	RSS_CLK_SRC_SEL		<a href="#">Section 5.2.1.68</a>
464h	PLLC_CLK2_SRC_SEL		<a href="#">Section 5.2.1.69</a>
468h	PLLD_CLK1_SRC_SEL		<a href="#">Section 5.2.1.70</a>
46Ch	PLLD_CLK2_SRC_SEL		<a href="#">Section 5.2.1.71</a>
470h	PLL_P_CLK1_SRC_SEL		<a href="#">Section 5.2.1.72</a>
474h	RSS_DIV_VAL		<a href="#">Section 5.2.1.73</a>
478h	RSS_CLK_GATE		<a href="#">Section 5.2.1.74</a>
47Ch	PLLC_CLK2_GATE		<a href="#">Section 5.2.1.75</a>
480h	PLLD_CLK1_GATE		<a href="#">Section 5.2.1.76</a>
484h	PLLD_CLK2_GATE		<a href="#">Section 5.2.1.77</a>
488h	PLL_P_CLK1_GATE		<a href="#">Section 5.2.1.78</a>
48Ch	RSS_CLK_STATUS		<a href="#">Section 5.2.1.79</a>
490h	PLLC_CLK2_STATUS		<a href="#">Section 5.2.1.80</a>
494h	PLLD_CLK1_STATUS		<a href="#">Section 5.2.1.81</a>
49Ch	PLL_P_CLK1_STATUS		<a href="#">Section 5.2.1.82</a>
4A0h	PLL_1P2_HSDIVIDER		<a href="#">Section 5.2.1.83</a>
4A4h	PLL_1P2_HSDIVIDER_CLKOUT0		<a href="#">Section 5.2.1.84</a>
4A8h	PLL_1P2_HSDIVIDER_CLKOUT1		<a href="#">Section 5.2.1.85</a>
4ACh	PLL_1P2_HSDIVIDER_CLKOUT2		<a href="#">Section 5.2.1.86</a>

**Table 5-2. MSS\_TOPRCM Registers (continued)**

Offset	Acronym	Register Name	Section
4B0h	PLL_1P2_HSDIVIDER_CLKOUT3		<a href="#">Section 5.2.1.87</a>
4B4h	PLL_1P2_HSDIVIDER_RSTCTRL		<a href="#">Section 5.2.1.88</a>
4B8h	PLL_1P8_HSDIVIDER		<a href="#">Section 5.2.1.89</a>
4BCh	PLL_1P8_HSDIVIDER_CLKOUT0		<a href="#">Section 5.2.1.90</a>
4C0h	PLL_1P8_HSDIVIDER_CLKOUT1		<a href="#">Section 5.2.1.91</a>
4C4h	PLL_1P8_HSDIVIDER_CLKOUT2		<a href="#">Section 5.2.1.92</a>
4C8h	PLL_1P8_HSDIVIDER_CLKOUT3		<a href="#">Section 5.2.1.93</a>
4CCh	PLL_1P8_HSDIVIDER_RSTCTRL		<a href="#">Section 5.2.1.94</a>
800h	PLL_DSP_PWRCTRL		<a href="#">Section 5.2.1.95</a>
804h	PLL_DSP_CLKCTRL		<a href="#">Section 5.2.1.96</a>
808h	PLL_DSP_TENABLE		<a href="#">Section 5.2.1.97</a>
80Ch	PLL_DSP_TENABLEDIV		<a href="#">Section 5.2.1.98</a>
810h	PLL_DSP_M2NDIV		<a href="#">Section 5.2.1.99</a>
814h	PLL_DSP_MN2DIV		<a href="#">Section 5.2.1.100</a>
818h	PLL_DSP_FRACDIV		<a href="#">Section 5.2.1.101</a>
81Ch	PLL_DSP_BWCTRL		<a href="#">Section 5.2.1.102</a>
820h	PLL_DSP_FRACCTRL		<a href="#">Section 5.2.1.103</a>
828h	PLL_DSP_HSDIVIDER		<a href="#">Section 5.2.1.104</a>
82Ch	PLL_DSP_HSDIVIDER_CLKOUT0		<a href="#">Section 5.2.1.105</a>
830h	PLL_DSP_HSDIVIDER_CLKOUT1		<a href="#">Section 5.2.1.106</a>
834h	PLL_DSP_HSDIVIDER_CLKOUT2		<a href="#">Section 5.2.1.107</a>
838h	PLL_DSP_HSDIVIDER_CLKOUT3		<a href="#">Section 5.2.1.108</a>
83Ch	PLL_PER_PWRCTRL		<a href="#">Section 5.2.1.109</a>
840h	PLL_PER_CLKCTRL		<a href="#">Section 5.2.1.110</a>
844h	PLL_PER_TENABLE		<a href="#">Section 5.2.1.111</a>
848h	PLL_PER_TENABLEDIV		<a href="#">Section 5.2.1.112</a>
84Ch	PLL_PER_M2NDIV		<a href="#">Section 5.2.1.113</a>
850h	PLL_PER_MN2DIV		<a href="#">Section 5.2.1.114</a>
854h	PLL_PER_FRACDIV		<a href="#">Section 5.2.1.115</a>
858h	PLL_PER_BWCTRL		<a href="#">Section 5.2.1.116</a>
85Ch	PLL_PER_FRACCTRL		<a href="#">Section 5.2.1.117</a>
860h	PLL_PER_STATUS		<a href="#">Section 5.2.1.118</a>
864h	PLL_PER_HSDIVIDER		<a href="#">Section 5.2.1.119</a>
868h	PLL_PER_HSDIVIDER_CLKOUT0		<a href="#">Section 5.2.1.120</a>
86Ch	PLL_PER_HSDIVIDER_CLKOUT1		<a href="#">Section 5.2.1.121</a>
870h	PLL_PER_HSDIVIDER_CLKOUT2		<a href="#">Section 5.2.1.122</a>
874h	PLL_PER_HSDIVIDER_CLKOUT3		<a href="#">Section 5.2.1.123</a>
878h	PLL_DSP_RSTCTRL		<a href="#">Section 5.2.1.124</a>
87Ch	PLL_DSP_HSDIVIDER_RSTCTRL		<a href="#">Section 5.2.1.125</a>
880h	PLL_PER_RSTCTRL		<a href="#">Section 5.2.1.126</a>
884h	PLL_PER_HSDIVIDER_RSTCTRL		<a href="#">Section 5.2.1.127</a>
C00h	ANA_REG_CLK_CTRL_REG1_XO_SLICE R		<a href="#">Section 5.2.1.128</a>
C04h	ANA_REG_CLK_CTRL_REG1_CLKTOP		<a href="#">Section 5.2.1.129</a>
C08h	ANA_REG_CLK_CTRL_REG2_CLKTOP		<a href="#">Section 5.2.1.130</a>

**Table 5-2. MSS\_TOPRCM Registers (continued)**

Offset	Acronym	Register Name	Section
C0Ch	ANA_REG_CLK_CTRL_REG1_LDO_CLK TOP		<a href="#">Section 5.2.1.131</a>
C10h	ANA_REG_CLK_CTRL_REG2_LDO_CLK TOP		<a href="#">Section 5.2.1.132</a>
C18h	ANA_REG_CLK_STATUS_REG		<a href="#">Section 5.2.1.133</a>
C1Ch	ANA_REG_REFSYS_CTRL_REG_LOWV		<a href="#">Section 5.2.1.134</a>
C20h	ANA_REG_REFSYS_TMUX_CTRL_LOW V		<a href="#">Section 5.2.1.135</a>
C24h	ANA_REG_REFSYS_SPARE_REG_LOW V		<a href="#">Section 5.2.1.136</a>
C28h	ANA_REG_WU_CTRL_REG_LOWV		<a href="#">Section 5.2.1.137</a>
C2Ch	ANA_REG_WU_TMUX_CTRL_LOWV		<a href="#">Section 5.2.1.138</a>
C30h	ANA_REG_TW_CTRL_REG_LOWV		<a href="#">Section 5.2.1.139</a>
C34h	ANA_REG_TW_ANA_TMUX_CTRL_LOW V		<a href="#">Section 5.2.1.140</a>
C3Ch	ANA_REG_WU_MODE_REG_LOWV		<a href="#">Section 5.2.1.141</a>
C40h	ANA_REG_WU_STATUS_REG_LOWV		<a href="#">Section 5.2.1.142</a>
C44h	ANA_REG_WU_SPARE_OUT_LOWV		<a href="#">Section 5.2.1.143</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.1.144</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.1.145</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.1.146</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.1.147</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.1.148</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.1.149</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.1.150</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.1.151</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.1.152</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.1.153</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.1.154</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.1.155</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.1.156</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.1.157</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.1.158</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.1.159</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.1.160</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.1.161</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.1.162</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.1.163</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.1.164</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-3](#) shows the codes that are used for access types in this section.

**Table 5-3. MSS\_TOPRCM Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read

**Table 5-3. MSS\_TOPRCM Access Type Codes  
(continued)**

Access Type	Code	Description
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.1.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-4](#).

Return to the [Summary Table](#).

PID register

**Table 5-4. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.1.2 HW\_REG0 Register (Offset = 4h) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-5](#).

Return to the [Summary Table](#).

**Table 5-5. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser. Reserved for HW RnD



### 5.2.1.3 HW\_REG1 Register (Offset = 8h) [Reset = 00000000h]

HW\_REG1 is shown in [Table 5-6](#).

Return to the [Summary Table](#).

**Table 5-6. HW\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved register. Reserved for HW RnD

#### 5.2.1.4 PREVIOUS\_NAME Register (Offset = Ch) [Reset = 00000000h]

PREVIOUS\_NAME is shown in [Table 5-7](#).

Return to the [Summary Table](#).

**Table 5-7. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser. Reserved for HW RnD

### 5.2.1.5 HW\_REG3 Register (Offset = 10h) [Reset = 00000000h]

HW\_REG3 is shown in [Table 5-8](#).

Return to the [Summary Table](#).

**Table 5-8. HW\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved register. Reserved for HW RnD

### 5.2.1.6 HSI\_CLK\_SRC\_SEL Register (Offset = 14h) [Reset = 00000000h]

HSI\_CLK\_SRC\_SEL is shown in [Table 5-9](#).

Return to the [Summary Table](#).

**Table 5-9. HSI\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for HSI. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.7 CSIRX\_CLK\_SRC\_SEL Register (Offset = 18h) [Reset = 00000000h]

CSIRX\_CLK\_SRC\_SEL is shown in [Table 5-10](#).

Return to the [Summary Table](#).

**Table 5-10. CSIRX\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for CSI Rx Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.8 MCUCLKOUT\_CLK\_SRC\_SEL Register (Offset = 1Ch) [Reset = 00000000h]

MCUCLKOUT\_CLK\_SRC\_SEL is shown in [Table 5-11](#).

Return to the [Summary Table](#).

**Table 5-11. MCUCLKOUT\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for MCU Clkout Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.9 PMICCLKOUT\_CLK\_SRC\_SEL Register (Offset = 20h) [Reset = 00000000h]

PMICCLKOUT\_CLK\_SRC\_SEL is shown in [Table 5-12](#).

Return to the [Summary Table](#).

**Table 5-12. PMICCLKOUT\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for PMIC Clkout Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.10 OBSCLKOUT\_CLK\_SRC\_SEL Register (Offset = 24h) [Reset = 00000000h]

OBSCLKOUT\_CLK\_SRC\_SEL is shown in [Table 5-13](#).

Return to the [Summary Table](#).

**Table 5-13. OBSCLKOUT\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for OBS Clkout Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference



### 5.2.1.11 TRCCLKOUT\_CLK\_SRC\_SEL Register (Offset = 28h) [Reset = 00000000h]

TRCCLKOUT\_CLK\_SRC\_SEL is shown in [Table 5-14](#).

Return to the [Summary Table](#).

**Table 5-14. TRCCLKOUT\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for TRC Clkout Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.12 CSIRX\_DIV\_VAL Register (Offset = 44h) [Reset = 00000000h]

CSIRX\_DIV\_VAL is shown in [Table 5-15](#).

Return to the [Summary Table](#).

**Table 5-15. CSIRX\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for CSI Rx selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.13 MCUCLKOUT\_DIV\_VAL Register (Offset = 48h) [Reset = 00000000h]

MCUCLKOUT\_DIV\_VAL is shown in [Table 5-16](#).

Return to the [Summary Table](#).

**Table 5-16. MCUCLKOUT\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for MCU Clkout selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.14 PMICCLKOUT\_DIV\_VAL Register (Offset = 4Ch) [Reset = 00000000h]

PMICCLKOUT\_DIV\_VAL is shown in [Table 5-17](#).

Return to the [Summary Table](#).

**Table 5-17. PMICCLKOUT\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for PMIC Clkout selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.15 OBSCLKOUT\_DIV\_VAL Register (Offset = 50h) [Reset = 00000000h]

OBSCLKOUT\_DIV\_VAL is shown in [Table 5-18](#).

Return to the [Summary Table](#).

**Table 5-18. OBSCLKOUT\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for OBS Clkout selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.16 TRCCLKOUT\_DIV\_VAL Register (Offset = 54h) [Reset = 00000000h]

TRCCLKOUT\_DIV\_VAL is shown in [Table 5-19](#).

Return to the [Summary Table](#).

**Table 5-19. TRCCLKOUT\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for TRC Clkout selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.17 CSIRX\_CLK\_GATE Register (Offset = 84h) [Reset = 00000000h]

CSIRX\_CLK\_GATE is shown in [Table 5-20](#).

Return to the [Summary Table](#).

**Table 5-20. CSIRX\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for CSI Rx. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.18 MCUCLKOUT\_CLK\_GATE Register (Offset = 88h) [Reset = 00000007h]

MCUCLKOUT\_CLK\_GATE is shown in [Table 5-21](#).

Return to the [Summary Table](#).

**Table 5-21. MCUCLKOUT\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	7h	Clock gating config for MCU Clkout. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)



### 5.2.1.19 PMICCLKOUT\_CLK\_GATE Register (Offset = 8Ch) [Reset = 00000007h]

PMICCLKOUT\_CLK\_GATE is shown in [Table 5-22](#).

Return to the [Summary Table](#).

**Table 5-22. PMICCLKOUT\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	7h	Clock gating config for PMIC Clkout Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.20 OBSCLKOUT\_CLK\_GATE Register (Offset = 90h) [Reset = 00000000h]

OBSCLKOUT\_CLK\_GATE is shown in [Table 5-23](#).

Return to the [Summary Table](#).

**Table 5-23. OBSCLKOUT\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for OBS Clkout Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.21 TRCCLKOUT\_CLK\_GATE Register (Offset = 94h) [Reset = 00000000h]

TRCCLKOUT\_CLK\_GATE is shown in [Table 5-24](#).

Return to the [Summary Table](#).

**Table 5-24. TRCCLKOUT\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for TRC Clkout Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.22 DSS\_CLK\_GATE Register (Offset = 98h) [Reset = 00000000h]

DSS\_CLK\_GATE is shown in [Table 5-25](#).

Return to the [Summary Table](#).

**Table 5-25. DSS\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for DSP Subsystem System Clock Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.23 HSI\_CLK\_STATUS Register (Offset = C0h) [Reset = 00000002h]

HSI\_CLK\_STATUS is shown in [Table 5-26](#).

Return to the [Summary Table](#).

**Table 5-26. HSI\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	clkinuse	R	2h	Status shows the source clock selected for CortexR5 Clock

### 5.2.1.24 MCUCLKOUT\_CLK\_STATUS Register (Offset = C8h) [Reset = 00000100h]

MCUCLKOUT\_CLK\_STATUS is shown in [Table 5-27](#).

Return to the [Summary Table](#).

**Table 5-27. MCUCLKOUT\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	1h	Status shows the current divider value choosen for CSI Rx Clock
7-0	clkinuse	R	0h	Status shows the source clock slected for CSI Rx Clock

### 5.2.1.25 PMICCLKOUT\_CLK\_STATUS Register (Offset = CCh) [Reset = 00000100h]

PMICCLKOUT\_CLK\_STATUS is shown in [Table 5-28](#).

Return to the [Summary Table](#).

**Table 5-28. PMICCLKOUT\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	1h	Status shows the current divider value choosen for MCU Clkout Clock
7-0	clkinuse	R	0h	Status shows the source clock slected for MCU Clkout Clock

### 5.2.1.26 OBSCLKOUT\_CLK\_STATUS Register (Offset = D0h) [Reset = 00000001h]

OBSCLKOUT\_CLK\_STATUS is shown in [Table 5-29](#).

Return to the [Summary Table](#).

**Table 5-29. OBSCLKOUT\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for PMIC Clkout Clock
7-0	clkinuse	R	1h	Status shows the source clock slected for PMIC Clkout Clock



### 5.2.1.27 TRCCLKOUT\_CLK\_STATUS Register (Offset = D4h) [Reset = 00000001h]

TRCCLKOUT\_CLK\_STATUS is shown in [Table 5-30](#).

Return to the [Summary Table](#).

**Table 5-30. TRCCLKOUT\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for PMIC Clkout Clock
7-0	clkinuse	R	1h	Status shows the source clock slected for PMIC Clkout Clock

### 5.2.1.28 WARM\_RESET\_CONFIG Register (Offset = 100h) [Reset = 0007XXXXh]

WARM\_RESET\_CONFIG is shown in [Table 5-31](#).

Return to the [Summary Table](#).

**Table 5-31. WARM\_RESET\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	wdog_rst_en	R/W	7h	Data should be loaded as multibit. Write 3'b000 to disable MSS Watchdog control on Warm reset Write 3'b111 enable MSS Watchdog to control Warm reset
15-11	RESERVED	R/W	0h	
10-8	sw_rst	R/W	7h	Data should be loaded as multibit. Write 3'b000 to assert warm reset from SW Write 3'b111 to deassert warm reset from SW if this is the only source of warm reset
7-3	RESERVED	R/W	0h	
2-0	pad_bypass	R/W	7h	Bypass the Warm reset from Pad Input Data should be loaded as multibit. Write 3'b000 : Reset is not asserted by SW (multibit 000) Write 3'b111 : Reset is asserted by SW (multibit 111)

### 5.2.1.29 SYS\_RST\_CAUSE Register (Offset = 104h) [Reset = 00000009h]

SYS\_RST\_CAUSE is shown in [Table 5-32](#).

Return to the [Summary Table](#).

**Table 5-32. SYS\_RST\_CAUSE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	cause	R	9h	System Reset Cause register 5'b01001 - POR reset 5'b01010 - Warm reset due to MSS_WDT 5'b01100 - Warm reset due to TOP_RMC:WARM_RESET_CONFIG 5'b01000 - External Pad reset 5'b11000 - Warm reset due to HSM_WDT

### 5.2.1.30 SYS\_RST\_CAUSE\_CLR Register (Offset = 108h) [Reset = 00000000h]

SYS\_RST\_CAUSE\_CLR is shown in [Table 5-33](#).

Return to the [Summary Table](#).

**Table 5-33. SYS\_RST\_CAUSE\_CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	clear	R/W	0h	Write pulse bit field: System Reset Cause register Clear

### 5.2.1.31 DSS\_RST\_CTRL Register (Offset = 10Ch) [Reset = 00000000h]

DSS\_RST\_CTRL is shown in [Table 5-34](#).

Return to the [Summary Table](#).

**Table 5-34. DSS\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	Reset control for DSP Subsystem Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW (multibit 000) Write 3'b111 : Reset is asserted by SW (multibit 111)

### 5.2.1.32 RS232\_BITINTERVAL Register (Offset = 204h) [Reset = 51615D5Bh]

RS232\_BITINTERVAL is shown in [Table 5-35](#).

Return to the [Summary Table](#).

**Table 5-35. RS232\_BITINTERVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	bitinterval	R/W	51615D5Bh	RS232 Bit Interval. 10 bit clock interval is selceted based on the value of MSS_CR5_CLK_SRC_SEL [9:0] used as RS232 Bit interval when MSS_CR5_CLK_SRC_SEL = 0x0 [19:10] used as RS232 Bit interval when MSS_CR5_CLK_SRC_SEL = 0x1 [29:20] used as RS232 Bit interval when MSS_CR5_CLK_SRC_SEL = 0x2

### 5.2.1.33 LVDS\_PAD\_CTRL0 Register (Offset = 208h) [Reset = 01010101h]

LVDS\_PAD\_CTRL0 is shown in [Table 5-36](#).

Return to the [Summary Table](#).

**Table 5-36. LVDS\_PAD\_CTRL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ctrl	R/W	01010101h	<p>LVDS Pad Control 0 Register. Below is the mapping for each bit. Refer the LVDS IO Spec for more details</p> <p>Bit 0 : Power Down Control for LVDS CLK Lane</p> <p>Bit 1: LOPWRA Control for i LVDS CLK Lane</p> <p>Bit 2: LOPWRB Control for LVDS CLK Lane</p> <p>Bit 3 : LPSEL Control for LVDS CLK Lane</p> <p>Bit 4 : SUB_LVDS_EN Control for LVDS CLK Lane</p> <p>Bit 5 : HIZ_DISABLE Control for LVDS CLK Lane</p> <p>Bit 6 : EXT_RES_EN Control for LVDS CLK Lane</p> <p>Bit 7 : Reserved</p> <p>Bit 8 : Power Down Control for LVDS DATA Lane 0</p> <p>Bit 9: LOPWRA Control for i LVDS DATA Lane 0</p> <p>Bit 10: LOPWRB Control for LVDS DATA Lane 0</p> <p>Bit 11: LPSEL Control for LVDS DATA Lane 0</p> <p>Bit 12: SUB_LVDS_EN Control for LVDS DATA Lane 0</p> <p>Bit 13: HIZ_DISABLE Control for LVDS DATA Lane 0</p> <p>Bit 14: EXT_RES_EN Control for LVDS DATA Lane 0</p> <p>Bit 15: Reserved</p> <p>Bit 16 : Power Down Control for LVDS DATA Lane 1</p> <p>Bit 17: LOPWRA Control for i LVDS DATA Lane 1</p> <p>Bit 18: LOPWRB Control for LVDS DATA Lane 1</p> <p>Bit 19: LPSEL Control for LVDS DATA Lane 1</p> <p>Bit 20: SUB_LVDS_EN Control for LVDS DATA Lane 1</p> <p>Bit 21: HIZ_DISABLE Control for LVDS DATA Lane 1</p> <p>Bit 22: EXT_RES_EN Control for LVDS DATA Lane 1</p> <p>Bit 23: Reserved</p> <p>Bit 24 : Power Down Control for LVDS DATA Lane 2</p> <p>Bit 25: LOPWRA Control for i LVDS DATA Lane 2</p> <p>Bit 26: LOPWRB Control for LVDS DATA Lane 2</p> <p>Bit 27: LPSEL Control for LVDS DATA Lane 2</p> <p>Bit 28: SUB_LVDS_EN Control for LVDS DATA Lane 2</p> <p>Bit 29: HIZ_DISABLE Control for LVDS DATA Lane 2</p> <p>Bit 30: EXT_RES_EN Control for LVDS DATA Lane 2</p> <p>Bit 31: Reserved</p>

### 5.2.1.34 LVDS\_PAD\_CTRL1 Register (Offset = 20Ch) [Reset = 00000101h]

LVDS\_PAD\_CTRL1 is shown in [Table 5-37](#).

Return to the [Summary Table](#).

**Table 5-37. LVDS\_PAD\_CTRL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ctrlr	R/W	101h	LVDS Pad Control 1 Register. Below is the mapping for each bit. Refer the LVDS IO Spec for more details Bit 0 : Power Down Control for LVDS DATA Lane 0 Bit 1: LOPWRA Control for i LVDS DATA Lane 0 Bit 2: LOPWRB Control for LVDS DATA Lane 0 Bit 3: LPSEL Control for LVDS DATA Lane 0 Bit 4: SUB_LVDS_EN Control for LVDS DATA Lane 0 Bit 5: HIZ_DISABLE Control for LVDS DATA Lane 0 Bit 6: EXT_RES_EN Control for LVDS DATA Lane 0 Bit 7: Reserved Bit 8 : Power Down Control for LVDS FRME CLK Lane Bit 9 : LOPWRA Control for i LVDS FRAME CLK Lane Bit 10: LOPWRB Control for LVDS FRAME CLK Lane Bit 11 : LPSEL Control for LVDS FRAME CLK Lane Bit 12 : SUB_LVDS_EN Control for LVDS FRAME CLK Lane Bit 13 : HIZ_DISABLE Control for LVDS FRAME CLK Lane Bit 14 : EXT_RES_EN Control for LVDS FRAME CLK Lane Bit 15 -23: Reserved Bit 24 : Power Down Control for LVDS Bias cell Bit 25 : eFuse Set Control for LVDS Bias cell



### 5.2.1.35 DFT\_DMLED\_EXEC Register (Offset = 210h) [Reset = 00000000h]

DFT\_DMLED\_EXEC is shown in [Table 5-38](#).

Return to the [Summary Table](#).

**Table 5-38. DFT\_DMLED\_EXEC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R/W	0h	SW mapping for DMLED Execution Bit 0 : HSM CM4 Execution Bit 1 : RESERVED Bit 2 : MSS CR5 Execution

### 5.2.1.36 DFT\_DMLED\_STATUS Register (Offset = 214h) [Reset = 00000000h]

DFT\_DMLED\_STATUS is shown in [Table 5-39](#).

Return to the [Summary Table](#).

**Table 5-39. DFT\_DMLED\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R/W	0h	SW mapping for DMLED Status Bit 0 : HSM CM4 Status Bit 1 : RESERVED Bit 2 : MSS CR5 Status

### 5.2.1.37 LIMP\_MODE\_EN Register (Offset = 218h) [Reset = 000000XXh]

LIMP\_MODE\_EN is shown in [Table 5-40](#).

Return to the [Summary Table](#).

**Table 5-40. LIMP\_MODE\_EN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	force_rcclk_en	R/W	0h	Force the RCCLK on when limp mode is detected 3'b000: The RCCLK will not be forced on when limp mode is detected (multibit 000) 3'b111 : The RCCLK will be forced on when limp mode is detected (multibit 111)
7	RESERVED	R/W	0h	
6-4	ccca_en	R/W	0h	Enable MSS_CCCA Error to generate Limp mode 3'b000: MSS_CCCA Error will not generate Limp mode (multibit 000) 3'b111 : MSS_CCCA Error will generate Limp mode (multibit 111)
3	RESERVED	R/W	0h	
2-0	dcca_en	R/W	0h	Enable MSS_DCCA Error to generate Limp mode 3'b000: MSS_DCCA Error will not generate Limp mode (multibit 000) 3'b111 : MSS_DCCA Error will generate Limp mode (multibit 111)

### 5.2.1.38 PMICCLKOUT\_DCDC\_CTRL Register (Offset = 21Ch) [Reset = 000000XXh]

PMICCLKOUT\_DCDC\_CTRL is shown in [Table 5-41](#).

Return to the [Summary Table](#).

**Table 5-41. PMICCLKOUT\_DCDC\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	max_freq_thr	R/W	0h	PMIC Clockout DCDC Maximum Frequency Threshold
15-8	min_freq_thr	R/W	0h	PMIC Clockout DCDC Minimum Frequency Threshold
7	RESERVED	R/W	0h	
6-4	reset_assert	R/W	0h	Reset control for PMIC DCDC Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW (multibit 000) Write 3'b111 : Reset is asserted by SW (multibit 111)
3	RESERVED	R/W	0h	
2	freq_acc_mode	R/W	0h	PMIC Clockout DCDC Freq Acc Enable
1	dither_en	R/W	0h	PMIC Clockout DCDC Clock Dither Enable
0	dcdc_clk_en	R/W	0h	PMIC Clockout DCDC Clock Enable

### 5.2.1.39 PMICCLKOUT\_DCDC\_SLOPE Register (Offset = 220h) [Reset = 00000000h]

PMICCLKOUT\_DCDC\_SLOPE is shown in [Table 5-42](#).

Return to the [Summary Table](#).

**Table 5-42. PMICCLKOUT\_DCDC\_SLOPE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-0	slope_val	R/W	0h	PMIC Clockout DCDC Slope Config Value

### 5.2.1.40 RCOSC32K\_CTRL Register (Offset = 224h) [Reset = 00000000h]

RCOSC32K\_CTRL is shown in [Table 5-43](#).

Return to the [Summary Table](#).

**Table 5-43. RCOSC32K\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	stoposc	R/W	0h	Stop 32KHz RCOSC. Write 3'b111 to stop clock

### 5.2.1.41 ANA\_HSI2DIGCLK\_GATE Register (Offset = 228h) [Reset = 00000007h]

ANA\_HSI2DIGCLK\_GATE is shown in [Table 5-44](#).

Return to the [Summary Table](#).

**Table 5-44. ANA\_HSI2DIGCLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	7h	Clock gating config for Analog HSI Clkout to Core GCms Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.42 DSI\_PHY\_PWRCTRL Register (Offset = 22Ch) [Reset = 00000033h]

DSI\_PHY\_PWRCTRL is shown in [Table 5-45](#).

Return to the [Summary Table](#).

**Table 5-45. DSI\_PHY\_PWRCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	PGOODOUT	R	1h	TI Reserved Dont touch.
4	PONOUT	R	1h	TI Reserved Dont touch.
3	ISOSCAN	R/W	0h	Save/Restore control for Isolation of the Scanout pins. For functional mode it should be 0
2	ISO	R/W	0h	Save/Restore control for Isolation of output pins For functional mode it should be 0
1	PGOODIN	R/W	1h	ON/OFF control of the strong power switch digital. For functional mode it should be 1
0	PONIN	R/W	1h	ON/OFF control of the weak power switch digital. For functional mode it should be 1



### 5.2.1.43 RS232\_SOP11\_BITINTERVAL Register (Offset = 230h) [Reset = 000001B2h]

RS232\_SOP11\_BITINTERVAL is shown in [Table 5-46](#).

Return to the [Summary Table](#).

**Table 5-46. RS232\_SOP11\_BITINTERVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R/W	1B2h	RS232 Bit Interval when SOP11 (ANA_WU_MODE_REG_LOWV[6:5]) is selected. XTAL is 50M and this field can be used to modify the baud rate.

#### 5.2.1.44 PLL\_CORE\_PWRCTRL Register (Offset = 400h) [Reset = 00000030h]

PLL\_CORE\_PWRCTRL is shown in [Table 5-47](#).

Return to the [Summary Table](#).

**Table 5-47. PLL\_CORE\_PWRCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	PONIN	R/W	1h	ON/OFF control of the weak power switch digital. For functional mode it should be 1
4	PGOODIN	R/W	1h	ON/OFF control of the strong power switch digital. For functional mode it should be 1
3	RET	R/W	0h	Save/Restore control for Retention mode. For functional mode it should be 0
2	ISORET	R/W	0h	Save/Restore control for Isolation of output pins For functional mode it should be 0
1	ISOSCAN	R/W	0h	Save/Restore control for Isolation of the Scanout pins. For functional mode it should be 0
0	OFFMODE	R/W	0h	Used to switch OFF the logic on VDDA. For functional mode it should be 0

### 5.2.1.45 PLL\_CORE\_CLKCTRL Register (Offset = 404h) [Reset = 0991XXXXh]

PLL\_CORE\_CLKCTRL is shown in [Table 5-48](#).

Return to the [Summary Table](#).

**Table 5-48. PLL\_CORE\_CLKCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CYCLESIPEN	R/W	0h	FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK.
30	ENSSC	R/W	0h	Controls Clock Spreading. SSC is not supported. Should be set to 0x0 to disable clock spreading.
29	CLKDCOLDOEN	R/W	0h	Synchronously enables/disables CLKDCOLDO 0x0 : synchronously disables CLKDCOLDO 0x1 : synchronously enables CLKDCOLDO
28-24	NWELLTRIM	R/W	9h	Trim value for the PLL
23	IDLE	R/W	1h	Sets PLL to Idle mode 0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked 0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power
22	BYPASSACKZ	R/W	0h	BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module.
21	STBYRET	R/W	0h	Standby retention control 0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks. 0x1 : prepares ADPLLLJ for retention by gating all the internal clocks.
20	CLKOUTEN	R/W	1h	CLKOUT enable or disable 0x0 : synchronously disables CLKOUT 0x1 : synchronously enables CLKOUT
19	CLKOUTLDOEN	R	0h	Synchronously enables/disables CLKOUTLDO 0x0 : synchronously disables CLKOUTLDO 0x1 : synchronously enables CLKOUTLDO
18	ULOWCLKEN	R/W	0h	Select CLKOUT source in bypass 0x0: When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1) 0x1: When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW.
17	CLKDCOLDOPWDNZ	R/W	0h	0 Asynchronous power down for CLKDCOLDO o/p.
16	M2PWDNZ	R/W	1h	M2 divider power down mode 0x0: Asynchronous power down for M2 divider 0x1 : M2 divider is functional
15	RESERVED	R/W	0h	
14	STOPMODE	R/W	1h	When in Lossclk/Stbyret 0x0 : Limp mode 0x1 : Stopmode
13	RESERVED	R/W	0h	
12-10	SELFREQDCO	R/W	2h	DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector. 0x0: Reserved 0x2: HS2 : DCOCLK range is from 500 MHz to 1000 MHz 0x3: Reserved 0x4: HS1: DCOCLK range is from 1000 MHz to 2000 MHz 0x5: Reserved
9	RESERVED	R/W	0h	
8	RELAXED_LOCK	R/W	0h	Decides when FREQLOCK asserted 0x0: FREQLOCK asserted when DC frequency error less than 1% 0x1: FREQLOCK asserted when DC frequency error less than 2%
7-2	RESERVED	R/W	0h	
1	SSCTYPE	R/W	0h	SSC Type
0	TINTZ	R/W	0h	PLL core soft reset

### 5.2.1.46 PLL\_CORE\_TENABLE Register (Offset = 408h) [Reset = 00000000h]

PLL\_CORE\_TENABLE is shown in [Table 5-49](#).

Return to the [Summary Table](#).

**Table 5-49. PLL\_CORE\_TENABLE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENABLE	R/W	0h	M, N, SD and SELFREQDCO latch (active rise edge)

### 5.2.1.47 PLL\_CORE\_TENABLEDIV Register (Offset = 40Ch) [Reset = 00000000h]

PLL\_CORE\_TENABLEDIV is shown in [Table 5-50](#).

Return to the [Summary Table](#).

**Table 5-50. PLL\_CORE\_TENABLEDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENABLEDIV	R/W	0h	M2 and N2 latch (active rise edge)

### 5.2.1.48 PLL\_CORE\_M2NDIV Register (Offset = 410h) [Reset = 0000XX00h]

PLL\_CORE\_M2NDIV is shown in [Table 5-51](#).

Return to the [Summary Table](#).

**Table 5-51. PLL\_CORE\_M2NDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	M2	R/W	0h	Post-divider is REGM2
15-8	RESERVED	R/W	0h	
7-0	N	R/W	0h	Pre-divider is REGN+1

### 5.2.1.49 PLL\_CORE\_MN2DIV Register (Offset = 414h) [Reset = 0000X174h]

PLL\_CORE\_MN2DIV is shown in [Table 5-52](#).

Return to the [Summary Table](#).

**Table 5-52. PLL\_CORE\_MN2DIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	N2	R/W	0h	Bypass divider is REGN2+1
15-12	RESERVED	R/W	0h	
11-0	M	R/W	174h	Feedback Multiplier is REGM

### 5.2.1.50 PLL\_CORE\_FRACDIV Register (Offset = 418h) [Reset = 08XX0000h]

PLL\_CORE\_FRACDIV is shown in [Table 5-53](#).

Return to the [Summary Table](#).

**Table 5-53. PLL\_CORE\_FRACDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	REGSD	R/W	8h	Sigma-Delta Divider Should be set by s/w to provide optimum jitter performance. $DPLL\_SD\_DIV = \text{CEILING} ([DPLL\_MULT / (DPLL\_DIV+1)] * CLKINP / 250)$ , where CLKINP is the input clock of the DPLL in MHz
23-18	RESERVED	R/W	0h	
17-0	FRACTIONALM	R/W	0h	Fractional part of the M divider.



### 5.2.1.51 PLL\_CORE\_BWCTRL Register (Offset = 41Ch) [Reset = 00000000h]

PLL\_CORE\_BWCTRL is shown in [Table 5-54](#).

Return to the [Summary Table](#).

**Table 5-54. PLL\_CORE\_BWCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-1	BWCONTROL	R/W	0h	Change Loop Bandwidth
0	BW_INCR_DECRZ	R/W	0h	Direction of Loop Bandwidth 0x0 : decrease BW 0x1 : increase BW

### 5.2.1.52 PLL\_CORE\_FRACCTRL Register (Offset = 420h) [Reset = 00000000h]

PLL\_CORE\_FRACCTRL is shown in [Table 5-55](#).

Return to the [Summary Table](#).

**Table 5-55. PLL\_CORE\_FRACCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	DOWNSPREAD	R/W	0h	Controls frequency spread 0x0 : enables both side frequency spread about the programmed frequency. 0x1 : enables low frequency spread only
30-28	ModFreqDividerExponent	R/W	0h	Exponent of the REFCLK divider to define the modulation frequency.
27-21	ModFreqDividerMantissa	R/W	0h	Mantissa of the REFCLK divider to define the modulation frequency
20-18	DeltaMStepInteger	R/W	0h	Integer part of Frequency Spread control
17-0	DeltaMStepFraction	R/W	0h	The fraction part of Frequency Spread control

### 5.2.1.53 PLL\_CORE\_STATUS Register (Offset = 424h) [Reset = EXXXX161h]

PLL\_CORE\_STATUS is shown in [Table 5-56](#).

Return to the [Summary Table](#).

**Table 5-56. PLL\_CORE\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PONOUT	R	1h	Status of the weak power-switch 0x0 : indicates the/OFF status of the weak power-switch in digital to SOC. 0x1 : ndicates the ON status of the weak power-switch in digital to SOC.
30	PGOODOUT	R	1h	Status of the strong power-switch 0x0 : indicates the/OFF status of the strong power-switch in digital to SOC. 0x1 : ndicates the ON status of the strong power-switch in digital to SOC.
29	LDOPWDN	R	1h	1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition
28	RECAL_BSTATUS3	R	0h	Recalibration status flag. 1 ADPLLLJ requires recalibration
27	RECAL_OPPIN	R	0h	Recalibration status flag. 1 ADPLLLJ requires recalibration
26-12	RESERVED	R	0h	
11	CLKDCOLDOACK	R	0h	1 indicates ACK 0 indicates no ACK
10	PHASELOCK	R	0h	1 indicates lock. 0 indicates no lock
9	FREQLOCK	R	0h	1 indicates lock. 0 indicates no lock
8	BYPASSACK	R	1h	1 indicates lock. 0 indicates no lock
7	STBYRETACK	R	0h	Standby and retention status 0x0: indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process. 0x1: indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention.
6	LOSSREF	R	1h	Reference input loss
5	CLKOUTENACK	R	1h	Indicates the enable/disable condition of CLKOUTEN 0x0 = CLKOUT gating completed 0x1 = CLKOUT enabling completed
4	LOCK2	R	0h	ADPLL internal loop lock status
3	M2CHANGEACK	R	0h	Acknowledge for change to M2 divider. Toggles from 1-0 or 0-1 (depending on current value) once CLKOUT frequency change has completed.
2	SSACK	R	0h	Spread Spectrum status 0x0 : Spread-spectrum Clocking is disabled on output clocks 0x1 : Spread-spectrum Clocking is enabled on output clocks
1	HIGHJITTER	R	0h	1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%.
0	BYPASS	R	1h	Bypass status signal. 1 CLKOUT in bypass

### 5.2.1.54 PLL\_CORE\_HSDIVIDER Register (Offset = 428h) [Reset = 0000XXXXh]

PLL\_CORE\_HSDIVIDER is shown in [Table 5-57](#).

Return to the [Summary Table](#).

**Table 5-57. PLL\_CORE\_HSDIVIDER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	LDOPWDNACK	R	0h	LDO Power Down Ack
16	BYPASSACKZ	R	0h	HSDIVIDER Bypass Ack
15-3	RESERVED	R/W	0h	
2	TENABLEDIV	R/W	0h	Tenable Div
1	LDOPWDN	R/W	0h	LDO Power Down
0	BYPASS	R/W	0h	HSDIVIDER Bypass

### 5.2.1.55 PLL\_CORE\_HSDIVIDER\_CLKOUT0 Register (Offset = 42Ch) [Reset = 0000XX4h]

PLL\_CORE\_HSDIVIDER\_CLKOUT0 is shown in [Table 5-58](#).

Return to the [Summary Table](#).

**Table 5-58. PLL\_CORE\_HSDIVIDER\_CLKOUT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M4 divider and hence CLKOUT0 output 0h (R/W) = CLKOUT0 divider active 1h (R/W) = CLKOUT0 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT0 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT0 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT0_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M4, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

### 5.2.1.56 PLL\_CORE\_HSDIVIDER\_CLKOUT1 Register (Offset = 430h) [Reset = 0000XX4h]

PLL\_CORE\_HSDIVIDER\_CLKOUT1 is shown in [Table 5-59](#).

Return to the [Summary Table](#).

**Table 5-59. PLL\_CORE\_HSDIVIDER\_CLKOUT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M5 divider and hence CLKOUT1 output 0h (R/W) = CLKOUT1 divider active 1h (R/W) = CLKOUT1 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT1 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT1 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M5, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

### 5.2.1.57 PLL\_CORE\_HSDIVIDER\_CLKOUT2 Register (Offset = 434h) [Reset = 0000XX4h]

PLL\_CORE\_HSDIVIDER\_CLKOUT2 is shown in [Table 5-60](#).

Return to the [Summary Table](#).

**Table 5-60. PLL\_CORE\_HSDIVIDER\_CLKOUT2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M6 divider and hence CLKOUT2 output 0h (R/W) = CLKOUT2 divider active 1h (R/W) = CLKOUT2 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT2 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT2 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M6, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

### 5.2.1.58 PLL\_CORE\_HSDIVIDER\_CLKOUT3 Register (Offset = 438h) [Reset = 0000XX4h]

PLL\_CORE\_HSDIVIDER\_CLKOUT3 is shown in [Table 5-61](#).

Return to the [Summary Table](#).

**Table 5-61. PLL\_CORE\_HSDIVIDER\_CLKOUT3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M7 divider and hence CLKOUT3 output 0h (R/W) = CLKOUT3 divider active 1h (R/W) = CLKOUT3 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT3 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT3 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M7, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved



### 5.2.1.59 MSS\_CR5\_CLK\_SRC\_SEL Register (Offset = 43Ch) [Reset = 00000000h]

MSS\_CR5\_CLK\_SRC\_SEL is shown in [Table 5-62](#).

Return to the [Summary Table](#).

**Table 5-62. MSS\_CR5\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for MSS Coretex R5 and System bus Clock. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.60 MSS\_CR5\_DIV\_VAL Register (Offset = 440h) [Reset = 00000000h]

MSS\_CR5\_DIV\_VAL is shown in [Table 5-63](#).

Return to the [Summary Table](#).

**Table 5-63. MSS\_CR5\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for Cortex R5 selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.61 SYS\_CLK\_DIV\_VAL Register (Offset = 444h) [Reset = 00000000h]

SYS\_CLK\_DIV\_VAL is shown in [Table 5-64](#).

Return to the [Summary Table](#).

**Table 5-64. SYS\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for System Clock selected clock. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.62 MSS\_CR5\_CLK\_GATE Register (Offset = 448h) [Reset = 00000000h]

MSS\_CR5\_CLK\_GATE is shown in [Table 5-65](#).

Return to the [Summary Table](#).

**Table 5-65. MSS\_CR5\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for MSS Cortex R5. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.63 SYS\_CLK\_GATE Register (Offset = 44Ch) [Reset = 00000000h]

SYS\_CLK\_GATE is shown in [Table 5-66](#).

Return to the [Summary Table](#).

**Table 5-66. SYS\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for System Clock Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.64 SYS\_CLK\_STATUS Register (Offset = 450h) [Reset = 000000XXh]

SYS\_CLK\_STATUS is shown in [Table 5-67](#).

Return to the [Summary Table](#).

**Table 5-67. SYS\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for Sys Clock
7-0	RESERVED	R	0h	

### 5.2.1.65 MSS\_CR5\_CLK\_STATUS Register (Offset = 454h) [Reset = 00000001h]

MSS\_CR5\_CLK\_STATUS is shown in [Table 5-68](#).

Return to the [Summary Table](#).

**Table 5-68. MSS\_CR5\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for CortexR5 Clock
7-0	clkinuse	R	1h	Status shows the source clock slected for CortexR5 Clock

### 5.2.1.66 PLL\_CORE\_RSTCTRL Register (Offset = 458h) [Reset = 00000000h]

PLL\_CORE\_RSTCTRL is shown in [Table 5-69](#).

Return to the [Summary Table](#).

**Table 5-69. PLL\_CORE\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the PLL Write 3'b111 : Override is enabled and Reset is asserted



### 5.2.1.67 PLL\_CORE\_HSDIVIDER\_RSTCTRL Register (Offset = 45Ch) [Reset = 00000000h]

PLL\_CORE\_HSDIVIDER\_RSTCTRL is shown in [Table 5-70](#).

Return to the [Summary Table](#).

**Table 5-70. PLL\_CORE\_HSDIVIDER\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the HSDIVIDER Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.68 RSS\_CLK\_SRC\_SEL Register (Offset = 460h) [Reset = 00000000h]

RSS\_CLK\_SRC\_SEL is shown in [Table 5-71](#).

Return to the [Summary Table](#).

**Table 5-71. RSS\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for RSS Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.69 PLLC\_CLK2\_SRC\_SEL Register (Offset = 464h) [Reset = 00000000h]

PLLC\_CLK2\_SRC\_SEL is shown in [Table 5-72](#).

Return to the [Summary Table](#).

**Table 5-72. PLLC\_CLK2\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for for PLLCORE_HSDIV_CLKOUT2_MUXED Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.70 PLLD\_CLK1\_SRC\_SEL Register (Offset = 468h) [Reset = 00000000h]

PLLD\_CLK1\_SRC\_SEL is shown in [Table 5-73](#).

Return to the [Summary Table](#).

**Table 5-73. PLLD\_CLK1\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for PLLCORE_HSDIV_CLKOUT2_MUXED Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.71 PLLD\_CLK2\_SRC\_SEL Register (Offset = 46Ch) [Reset = 00000000h]

PLLD\_CLK2\_SRC\_SEL is shown in [Table 5-74](#).

Return to the [Summary Table](#).

**Table 5-74. PLLD\_CLK2\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for PLLDSP_HSDIV_CLKOUT2_MUXED. Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.72 PLLP\_CLK1\_SRC\_SEL Register (Offset = 470h) [Reset = 00000000h]

PLLP\_CLK1\_SRC\_SEL is shown in [Table 5-75](#).

Return to the [Summary Table](#).

**Table 5-75. PLLP\_CLK1\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for PLLPER_HSDIV_CLKOUT1_MUXED Data should be loaded as multibit. For example: if Clock source 0x5 should be selected then 0x555 should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.1.73 RSS\_DIV\_VAL Register (Offset = 474h) [Reset = 00000000h]

RSS\_DIV\_VAL is shown in [Table 5-76](#).

Return to the [Summary Table](#).

**Table 5-76. RSS\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdiv	R/W	0h	Divider value for RSS. Data should be loaded as multibit. For example: if divider value of 0x5 should be selected then 0x555 should be configured to the register.

### 5.2.1.74 RSS\_CLK\_GATE Register (Offset = 478h) [Reset = 00000000h]

RSS\_CLK\_GATE is shown in [Table 5-77](#).

Return to the [Summary Table](#).

**Table 5-77. RSS\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for RSS. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)



### 5.2.1.75 PLLC\_CLK2\_GATE Register (Offset = 47Ch) [Reset = 00000000h]

PLLC\_CLK2\_GATE is shown in [Table 5-78](#).

Return to the [Summary Table](#).

**Table 5-78. PLLC\_CLK2\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for PLLCORE_HSDIV_CLKOUT2_MUXED. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.76 PLLD\_CLK1\_GATE Register (Offset = 480h) [Reset = 00000000h]

PLLD\_CLK1\_GATE is shown in [Table 5-79](#).

Return to the [Summary Table](#).

**Table 5-79. PLLD\_CLK1\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for PLLCORE_HSDIV_CLKOUT2_MUXED.. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.77 PLLD\_CLK2\_GATE Register (Offset = 484h) [Reset = 00000000h]

PLLD\_CLK2\_GATE is shown in [Table 5-80](#).

Return to the [Summary Table](#).

**Table 5-80. PLLD\_CLK2\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for PLLDSP_HSDIV_CLKOUT2_MUXED. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.78 PLLP\_CLK1\_GATE Register (Offset = 488h) [Reset = 00000000h]

PLLP\_CLK1\_GATE is shown in [Table 5-81](#).

Return to the [Summary Table](#).

**Table 5-81. PLLP\_CLK1\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for PLLPER_HSDIV_CLKOUT1_MUXED. Data should be loaded as multibit. Write 3'b000 : Clock is ungated (multibit 000) Write 3'b111 : Clock is gated (multibit 111)

### 5.2.1.79 RSS\_CLK\_STATUS Register (Offset = 48Ch) [Reset = 00000001h]

RSS\_CLK\_STATUS is shown in [Table 5-82](#).

Return to the [Summary Table](#).

**Table 5-82. RSS\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for RSS
7-0	clkinuse	R	1h	Status shows the source clock slected for RSS

### 5.2.1.80 PLLC\_CLK2\_STATUS Register (Offset = 490h) [Reset = 00000001h]

PLLC\_CLK2\_STATUS is shown in [Table 5-83](#).

Return to the [Summary Table](#).

**Table 5-83. PLLC\_CLK2\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	clkinuse	R	1h	Status shows the source clock selected for GCM switch for PLLCORE_HSDIV_CLK2

### 5.2.1.81 PLLD\_CLK1\_STATUS Register (Offset = 494h) [Reset = 00000001h]

PLLD\_CLK1\_STATUS is shown in [Table 5-84](#).

Return to the [Summary Table](#).

**Table 5-84. PLLD\_CLK1\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	clkinuse	R	1h	Status shows the source clock selected for GCM switch for PLLCORE_HSDIV_CLKOUT2_MUXED

### 5.2.1.82 PLLP\_CLK1\_STATUS Register (Offset = 49Ch) [Reset = 00000001h]

PLLP\_CLK1\_STATUS is shown in [Table 5-85](#).

Return to the [Summary Table](#).

**Table 5-85. PLLP\_CLK1\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	clkinuse	R	1h	Status shows the source clock selected for GCM switch for PLLPER_HSDIV_CLK1



### 5.2.1.83 PLL\_1P2\_HSDIVIDER Register (Offset = 4A0h) [Reset = 0000XXXXh]

PLL\_1P2\_HSDIVIDER is shown in [Table 5-86](#).

Return to the [Summary Table](#).

**Table 5-86. PLL\_1P2\_HSDIVIDER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	LDOPWDNACK	R	0h	Do not use. TI Reserved.
16	BYPASSACKZ	R	0h	Do not use. TI Reserved.
15-3	RESERVED	R/W	0h	
2	TENABLEDIV	R/W	0h	Do not use. TI Reserved.
1	LDOPWDN	R/W	0h	Do not use. TI Reserved.
0	BYPASS	R/W	0h	Do not use. TI Reserved.

### 5.2.1.84 PLL\_1P2\_HSDIVIDER\_CLKOUT0 Register (Offset = 4A4h) [Reset = 00000XX4h]

PLL\_1P2\_HSDIVIDER\_CLKOUT0 is shown in [Table 5-87](#).

Return to the [Summary Table](#).

**Table 5-87. PLL\_1P2\_HSDIVIDER\_CLKOUT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.85 PLL\_1P2\_HSDIVIDER\_CLKOUT1 Register (Offset = 4A8h) [Reset = 00000XX4h]

PLL\_1P2\_HSDIVIDER\_CLKOUT1 is shown in [Table 5-88](#).

Return to the [Summary Table](#).

**Table 5-88. PLL\_1P2\_HSDIVIDER\_CLKOUT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.86 PLL\_1P2\_HSDIVIDER\_CLKOUT2 Register (Offset = 4ACh) [Reset = 00000XX4h]

PLL\_1P2\_HSDIVIDER\_CLKOUT2 is shown in [Table 5-89](#).

Return to the [Summary Table](#).

**Table 5-89. PLL\_1P2\_HSDIVIDER\_CLKOUT2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.87 PLL\_1P2\_HSDIVIDER\_CLKOUT3 Register (Offset = 4B0h) [Reset = 00000XX4h]

PLL\_1P2\_HSDIVIDER\_CLKOUT3 is shown in [Table 5-90](#).

Return to the [Summary Table](#).

**Table 5-90. PLL\_1P2\_HSDIVIDER\_CLKOUT3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.88 PLL\_1P2\_HSDIVIDER\_RSTCTRL Register (Offset = 4B4h) [Reset = 00000000h]

PLL\_1P2\_HSDIVIDER\_RSTCTRL is shown in [Table 5-91](#).

Return to the [Summary Table](#).

**Table 5-91. PLL\_1P2\_HSDIVIDER\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the HSDIVIDER Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.89 PLL\_1P8\_HSDIVIDER Register (Offset = 4B8h) [Reset = 0000XXXh]

PLL\_1P8\_HSDIVIDER is shown in [Table 5-92](#).

Return to the [Summary Table](#).

**Table 5-92. PLL\_1P8\_HSDIVIDER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	LDOPWDNACK	R	0h	Do not use. TI Reserved.
16	BYPASSACKZ	R	0h	Do not use. TI Reserved.
15-3	RESERVED	R/W	0h	
2	TENABLEDIV	R/W	0h	Do not use. TI Reserved.
1	LDOPWDN	R/W	0h	Do not use. TI Reserved.
0	BYPASS	R/W	0h	Do not use. TI Reserved.

### 5.2.1.90 PLL\_1P8\_HSDIVIDER\_CLKOUT0 Register (Offset = 4BCh) [Reset = 00000XX4h]

PLL\_1P8\_HSDIVIDER\_CLKOUT0 is shown in [Table 5-93](#).

Return to the [Summary Table](#).

**Table 5-93. PLL\_1P8\_HSDIVIDER\_CLKOUT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.



### 5.2.1.91 PLL\_1P8\_HSDIVIDER\_CLKOUT1 Register (Offset = 4C0h) [Reset = 00000XX4h]

PLL\_1P8\_HSDIVIDER\_CLKOUT1 is shown in [Table 5-94](#).

Return to the [Summary Table](#).

**Table 5-94. PLL\_1P8\_HSDIVIDER\_CLKOUT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.92 PLL\_1P8\_HSDIVIDER\_CLKOUT2 Register (Offset = 4C4h) [Reset = 00000XX4h]

PLL\_1P8\_HSDIVIDER\_CLKOUT2 is shown in [Table 5-95](#).

Return to the [Summary Table](#).

**Table 5-95. PLL\_1P8\_HSDIVIDER\_CLKOUT2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.93 PLL\_1P8\_HSDIVIDER\_CLKOUT3 Register (Offset = 4C8h) [Reset = 00000XX4h]

PLL\_1P8\_HSDIVIDER\_CLKOUT3 is shown in [Table 5-96](#).

Return to the [Summary Table](#).

**Table 5-96. PLL\_1P8\_HSDIVIDER\_CLKOUT3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.94 PLL\_1P8\_HSDIVIDER\_RSTCTRL Register (Offset = 4CCh) [Reset = 00000000h]

PLL\_1P8\_HSDIVIDER\_RSTCTRL is shown in [Table 5-97](#).

Return to the [Summary Table](#).

**Table 5-97. PLL\_1P8\_HSDIVIDER\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the HSDIVIDER Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.95 PLL\_DSP\_PWRCTRL Register (Offset = 800h) [Reset = 00000030h]

PLL\_DSP\_PWRCTRL is shown in [Table 5-98](#).

Return to the [Summary Table](#).

**Table 5-98. PLL\_DSP\_PWRCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	PONIN	R/W	1h	Do not use. TI Reserved.
4	PGOODIN	R/W	1h	Do not use. TI Reserved.
3	RET	R/W	0h	Do not use. TI Reserved.
2	ISORET	R/W	0h	Do not use. TI Reserved.
1	ISOSCAN	R/W	0h	Do not use. TI Reserved.
0	OFFMODE	R/W	0h	Do not use. TI Reserved.

### 5.2.1.96 PLL\_DSP\_CLKCTRL Register (Offset = 804h) [Reset = 0991XXXXh]

PLL\_DSP\_CLKCTRL is shown in [Table 5-99](#).

Return to the [Summary Table](#).

**Table 5-99. PLL\_DSP\_CLKCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CYCLESIPEN	R/W	0h	Do not use. TI Reserved.
30	ENSSC	R/W	0h	Do not use. TI Reserved.
29	CLKDCOLDOEN	R/W	0h	Do not use. TI Reserved.
28-24	NWELLTRIM	R/W	9h	Do not use. TI Reserved.
23	IDLE	R/W	1h	Do not use. TI Reserved.
22	BYPASSACKZ	R/W	0h	Do not use. TI Reserved.
21	STBYRET	R/W	0h	Do not use. TI Reserved.
20	CLKOUTEN	R/W	1h	Do not use. TI Reserved.
19	CLKOUTLDOEN	R	0h	Do not use. TI Reserved.
18	ULOWCLKEN	R/W	0h	Do not use. TI Reserved.
17	CLKDCOLDOPWDNZ	R/W	0h	Do not use. TI Reserved.
16	M2PWDNZ	R/W	1h	Do not use. TI Reserved.
15	RESERVED	R/W	0h	
14	STOPMODE	R/W	1h	Do not use. TI Reserved.
13	RESERVED	R/W	0h	
12-10	SELFREQDCO	R/W	2h	Do not use. TI Reserved.
9	RESERVED	R/W	0h	
8	RELAXED_LOCK	R/W	0h	Do not use. TI Reserved.
7-2	RESERVED	R/W	0h	
1	SSCTYPE	R/W	0h	Do not use. TI Reserved.
0	TINTZ	R/W	0h	Do not use. TI Reserved.

### 5.2.1.97 PLL\_DSP\_TENABLE Register (Offset = 808h) [Reset = 00000000h]

PLL\_DSP\_TENABLE is shown in [Table 5-100](#).

Return to the [Summary Table](#).

**Table 5-100. PLL\_DSP\_TENABLE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENABLE	R/W	0h	Do not use. TI Reserved.

### 5.2.1.98 PLL\_DSP\_TENABLEDIV Register (Offset = 80Ch) [Reset = 00000000h]

PLL\_DSP\_TENABLEDIV is shown in [Table 5-101](#).

Return to the [Summary Table](#).

**Table 5-101. PLL\_DSP\_TENABLEDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENABLEDIV	R/W	0h	Do not use. TI Reserved.



### 5.2.1.99 PLL\_DSP\_M2NDIV Register (Offset = 810h) [Reset = 0000XX00h]

PLL\_DSP\_M2NDIV is shown in [Table 5-102](#).

Return to the [Summary Table](#).

**Table 5-102. PLL\_DSP\_M2NDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	M2	R/W	0h	Do not use. TI Reserved.
15-8	RESERVED	R/W	0h	
7-0	N	R/W	0h	Do not use. TI Reserved.

### 5.2.1.100 PLL\_DSP\_MN2DIV Register (Offset = 814h) [Reset = 0000X174h]

PLL\_DSP\_MN2DIV is shown in [Table 5-103](#).

Return to the [Summary Table](#).

**Table 5-103. PLL\_DSP\_MN2DIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	N2	R/W	0h	Do not use. TI Reserved.
15-12	RESERVED	R/W	0h	
11-0	M	R/W	174h	Do not use. TI Reserved.

### 5.2.1.101 PLL\_DSP\_FRACDIV Register (Offset = 818h) [Reset = 08XX0000h]

PLL\_DSP\_FRACDIV is shown in [Table 5-104](#).

Return to the [Summary Table](#).

**Table 5-104. PLL\_DSP\_FRACDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	REGSD	R/W	8h	Do not use. TI Reserved.
23-18	RESERVED	R/W	0h	
17-0	FRACTIONALM	R/W	0h	Do not use. TI Reserved.

### 5.2.1.102 PLL\_DSP\_BWCTRL Register (Offset = 81Ch) [Reset = 00000000h]

PLL\_DSP\_BWCTRL is shown in [Table 5-105](#).

Return to the [Summary Table](#).

**Table 5-105. PLL\_DSP\_BWCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-1	BWCONTROL	R/W	0h	Do not use. TI Reserved.
0	BW_INCR_DECRZ	R/W	0h	Do not use. TI Reserved.

### 5.2.1.103 PLL\_DSP\_FRACCTRL Register (Offset = 820h) [Reset = 00000000h]

PLL\_DSP\_FRACCTRL is shown in [Table 5-106](#).

Return to the [Summary Table](#).

**Table 5-106. PLL\_DSP\_FRACCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	DOWNSPREAD	R/W	0h	Do not use. TI Reserved.
30-28	ModFreqDividerExponent	R/W	0h	Do not use. TI Reserved.
27-21	ModFreqDividerMantissa	R/W	0h	Do not use. TI Reserved.
20-18	DeltaMStepInteger	R/W	0h	Do not use. TI Reserved.
17-0	DeltaMStepFraction	R/W	0h	Do not use. TI Reserved.

### 5.2.1.104 PLL\_DSP\_HSDIVIDER Register (Offset = 828h) [Reset = 0000XXXXh]

PLL\_DSP\_HSDIVIDER is shown in [Table 5-107](#).

Return to the [Summary Table](#).

**Table 5-107. PLL\_DSP\_HSDIVIDER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	LDOPWDNACK	R	0h	Do not use. TI Reserved.
16	BYPASSACKZ	R	0h	Do not use. TI Reserved.
15-3	RESERVED	R/W	0h	
2	TENABLEDIV	R/W	0h	Do not use. TI Reserved.
1	LDOPWDN	R/W	0h	Do not use. TI Reserved.
0	BYPASS	R/W	0h	Do not use. TI Reserved.

### 5.2.1.105 PLL\_DSP\_HSDIVIDER\_CLKOUT0 Register (Offset = 82Ch) [Reset = 00000XX4h]

PLL\_DSP\_HSDIVIDER\_CLKOUT0 is shown in [Table 5-108](#).

Return to the [Summary Table](#).

**Table 5-108. PLL\_DSP\_HSDIVIDER\_CLKOUT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.106 PLL\_DSP\_HSDIVIDER\_CLKOUT1 Register (Offset = 830h) [Reset = 00000XX4h]

PLL\_DSP\_HSDIVIDER\_CLKOUT1 is shown in [Table 5-109](#).

Return to the [Summary Table](#).

**Table 5-109. PLL\_DSP\_HSDIVIDER\_CLKOUT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.



### 5.2.1.107 PLL\_DSP\_HSDIVIDER\_CLKOUT2 Register (Offset = 834h) [Reset = 00000XX4h]

PLL\_DSP\_HSDIVIDER\_CLKOUT2 is shown in [Table 5-110](#).

Return to the [Summary Table](#).

**Table 5-110. PLL\_DSP\_HSDIVIDER\_CLKOUT2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.108 PLL\_DSP\_HSDIVIDER\_CLKOUT3 Register (Offset = 838h) [Reset = 00000XX4h]

PLL\_DSP\_HSDIVIDER\_CLKOUT3 is shown in [Table 5-111](#).

Return to the [Summary Table](#).

**Table 5-111. PLL\_DSP\_HSDIVIDER\_CLKOUT3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Do not use. TI Reserved.
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	Do not use. TI Reserved.
8	GATE_CTRL	R/W	0h	Do not use. TI Reserved.
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Do not use. TI Reserved.
4-0	DIV	R/W	4h	Do not use. TI Reserved.

### 5.2.1.109 PLL\_PER\_PWRCTRL Register (Offset = 83Ch) [Reset = 00000030h]

PLL\_PER\_PWRCTRL is shown in [Table 5-112](#).

Return to the [Summary Table](#).

**Table 5-112. PLL\_PER\_PWRCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5	PONIN	R/W	1h	ON/OFF control of the weak power switch digital. For functional mode it should be 1
4	PGOODIN	R/W	1h	ON/OFF control of the strong power switch digital. For functional mode it should be 1
3	RET	R/W	0h	Save/Restore control for Retention mode. For functional mode it should be 0
2	ISORET	R/W	0h	Save/Restore control for Isolation of output pins For functional mode it should be 0
1	ISOSCAN	R/W	0h	Save/Restore control for Isolation of the Scanout pins. For functional mode it should be 0
0	OFFMODE	R/W	0h	Used to switch OFF the logic on VDDA. For functional mode it should be 0

### 5.2.1.110 PLL\_PER\_CLKCTRL Register (Offset = 840h) [Reset = 0991XXXXh]

PLL\_PER\_CLKCTRL is shown in [Table 5-113](#).

Return to the [Summary Table](#).

**Table 5-113. PLL\_PER\_CLKCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CYCLESIPEN	R/W	0h	FailSafe enable to trigger re-calibration in case CycleSlip occurs between REFCLK and FBCLK.
30	ENSSC	R/W	0h	Controls Clock Spreading. SSC is not supported. Should be set to 0x0 to disable clock spreading.
29	CLKDCOLDOEN	R/W	0h	Synchronously enables/disables CLKDCOLDO 0x0 : synchronously disables CLKDCOLDO 0x1 : synchronously enables CLKDCOLDO
28-24	NWELLTRIM	R/W	9h	Trim value for the PLL
23	IDLE	R/W	1h	Sets PLL to Idle mode 0x0 : When SYSRESET = 0 and TINITZ = 1 IDLE = 0 PLL will go to Active and Locked 0x1 : When SYSRESET = 0 and TINITZ = 1 IDLE = 1 PLL will go to Idle Bypass low power
22	BYPASSACKZ	R/W	0h	BYPASSACKZ is a special purpose input to the module. In general this input is expected to be tied to static low. For the output clocks of the module that do not have an internal bypass mux viz. CLKDCOLDO and CLKOUTLDO, a bypass mux could be implemented external to the module.
21	STBYRET	R/W	0h	Standby retention control 0x0 : prepares ADPLLLJ for relock when out of retention by removing the gating on all internal clocks. 0x1 : prepares ADPLLLJ for retention by gating all the internal clocks.
20	CLKOUTEN	R/W	1h	CLKOUT enable or disable 0x0 : synchronously disables CLKOUT 0x1 : synchronously enables CLKOUT
19	CLKOUTLDOEN	R	0h	Synchronously enables/disables CLKOUTLDO 0x0 : synchronously disables CLKOUTLDO 0x1 : synchronously enables CLKOUTLDO
18	ULOWCLKEN	R/W	0h	Select CLKOUT source in bypass 0x0: When ADPLLLJ in bypass mode, CLKOUT = CLKINP/(N2+1) 0x1: When ADPLLLJ in bypass mode, CLKOUT = CLKINPULOW.
17	CLKDCOLDOPWDNZ	R/W	0h	0 Asynchronous power down for CLKDCOLDO o/p.
16	M2PWDNZ	R/W	1h	M2 divider power down mode 0x0: Asynchronous power down for M2 divider 0x1 : M2 divider is functional
15	RESERVED	R/W	0h	
14	STOPMODE	R/W	1h	When in Lossclk/Stbyret 0x0 : Limp mode 0x1 : Stopmode
13	RESERVED	R/W	0h	
12-10	SELFREQDCO	R/W	2h	DCO Clock (DCOCLK = CLKINP * [M/(N+1)]) frequency range selector. 0x0: Reserved 0x2: HS2 : DCOCLK range is from 500 MHz to 1000 MHz 0x3: Reserved 0x4: HS1: DCOCLK range is from 1000 MHz to 2000 MHz 0x5: Reserved
9	RESERVED	R/W	0h	
8	RELAXED_LOCK	R/W	0h	Decides when FREQLOCK asserted 0x0: FREQLOCK asserted when DC frequency error less than 1% 0x1: FREQLOCK asserted when DC frequency error less than 2%
7-2	RESERVED	R/W	0h	
1	SSCTYPE	R/W	0h	SSC Type
0	TINTZ	R/W	0h	PLL core soft reset

### 5.2.1.111 PLL\_PER\_TENABLE Register (Offset = 844h) [Reset = 00000000h]

PLL\_PER\_TENABLE is shown in [Table 5-114](#).

Return to the [Summary Table](#).

**Table 5-114. PLL\_PER\_TENABLE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENABLE	R/W	0h	M, N, SD and SELFREQDCO latch (active rise edge)

### 5.2.1.112 PLL\_PER\_TENABLEDIV Register (Offset = 848h) [Reset = 00000000h]

PLL\_PER\_TENABLEDIV is shown in [Table 5-115](#).

Return to the [Summary Table](#).

**Table 5-115. PLL\_PER\_TENABLEDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	TENABLEDIV	R/W	0h	M2 and N2 latch (active rise edge)

### 5.2.1.113 PLL\_PER\_M2NDIV Register (Offset = 84Ch) [Reset = 0000XX00h]

PLL\_PER\_M2NDIV is shown in [Table 5-116](#).

Return to the [Summary Table](#).

**Table 5-116. PLL\_PER\_M2NDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-16	M2	R/W	0h	Post-divider is REGM2
15-8	RESERVED	R/W	0h	
7-0	N	R/W	0h	Pre-divider is REGN+1

### 5.2.1.114 PLL\_PER\_MN2DIV Register (Offset = 850h) [Reset = 0000X174h]

PLL\_PER\_MN2DIV is shown in [Table 5-117](#).

Return to the [Summary Table](#).

**Table 5-117. PLL\_PER\_MN2DIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-16	N2	R/W	0h	Bypass divider is REGN2+1
15-12	RESERVED	R/W	0h	
11-0	M	R/W	174h	Feedback Multiplier is REGM



### 5.2.1.115 PLL\_PER\_FRACDIV Register (Offset = 854h) [Reset = 08XX0000h]

PLL\_PER\_FRACDIV is shown in [Table 5-118](#).

Return to the [Summary Table](#).

**Table 5-118. PLL\_PER\_FRACDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	REGSD	R/W	8h	Sigma-Delta Divider Should be set by s/w to provide optimum jitter performance. $DPLL\_SD\_DIV = \text{CEILING} ([DPLL\_MULT / (DPLL\_DIV+1)] * CLKINP / 250)$ , where CLKINP is the input clock of the DPLL in MHz
23-18	RESERVED	R/W	0h	
17-0	FRACTIONALM	R/W	0h	Fractional part of the M divider.

### 5.2.1.116 PLL\_PER\_BWCTRL Register (Offset = 858h) [Reset = 00000000h]

PLL\_PER\_BWCTRL is shown in [Table 5-119](#).

Return to the [Summary Table](#).

**Table 5-119. PLL\_PER\_BWCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-1	BWCONTROL	R/W	0h	Change Loop Bandwidth
0	BW_INCR_DECRZ	R/W	0h	Direction of Loop Bandwidth 0x0 : decrease BW 0x1 : increase BW

### 5.2.1.117 PLL\_PER\_FRACCTRL Register (Offset = 85Ch) [Reset = 00000000h]

PLL\_PER\_FRACCTRL is shown in [Table 5-120](#).

Return to the [Summary Table](#).

**Table 5-120. PLL\_PER\_FRACCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	DOWNSPREAD	R/W	0h	Controls frequency spread 0x0 : enables both side frequency spread about the programmed frequency. 0x1 : enables low frequency spread only
30-28	ModFreqDividerExponent	R/W	0h	Exponent of the REFCLK divider to define the modulation frequency.
27-21	ModFreqDividerMantissa	R/W	0h	Mantissa of the REFCLK divider to define the modulation frequency
20-18	DeltaMStepInteger	R/W	0h	Integer part of Frequency Spread control
17-0	DeltaMStepFraction	R/W	0h	The fraction part of Frequency Spread control

### 5.2.1.118 PLL\_PER\_STATUS Register (Offset = 860h) [Reset = EXXX161h]

PLL\_PER\_STATUS is shown in [Table 5-121](#).

Return to the [Summary Table](#).

**Table 5-121. PLL\_PER\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PONOUT	R	1h	Status of the weak power-switch 0x0 : indicates the/OFF status of the weak power-switch in digital to SOC. 0x1 : ndicates the ON status of the weak power-switch in digital to SOC.
30	PGOODOUT	R	1h	Status of the strong power-switch 0x0 : indicates the/OFF status of the strong power-switch in digital to SOC. 0x1 : ndicates the ON status of the strong power-switch in digital to SOC.
29	LDOPWDN	R	1h	1 indicates ADPLLLJ internal LDO is power down. VDDLDOOUT will be un-defined in this condition
28	RECAL_BSTATUS3	R	0h	Recalibration status flag. 1 ADPLLLJ requires recalibration
27	RECAL_OPPIN	R	0h	Recalibration status flag. 1 ADPLLLJ requires recalibration
26-12	RESERVED	R	0h	
11	CLKDCOLDOACK	R	0h	Status on PHASELOCK output pin
10	PHASELOCK	R	0h	Status on PHASELOCK output pin
9	FREQLOCK	R	0h	Status on FREQLOCK output pin
8	BYPASSACK	R	1h	Status of BYPASSACK output pin
7	STBYRETACK	R	0h	Standby and retention status 0x0: indicates to SOC that all internal clocks in ADPLLLJ are active and it is starting the relock process. 0x1: indicates to SOC that all internal clocks in ADPLLLJ are gated and it is ready for retention.
6	LOSSREF	R	1h	Reference input loss
5	CLKOUTENACK	R	1h	Indicates the enable/disable condition of CLKOUTEN 0x0 = CLKOUT gating completed 0x1 = CLKOUT enabling completed
4	LOCK2	R	0h	ADPLL internal loop lock status
3	M2CHANGEACK	R	0h	Acknowledge for change to M2 divider. Toggles from 1-0 or 0-1 (depending on current value) once CLKOUT frequency change has completed.
2	SSACK	R	0h	Spread Spectrum status 0x0 : Spread-spectrum Clocking is disabled on output clocks 0x1 : Spread-spectrum Clocking is enabled on output clocks
1	HIGHJITTER	R	0h	1 indicates jitter. After PHASELOCK is asserted high, the HIGHJITTER flag is asserted high if phase error between REFCLK and FBCLK greater than 24%.
0	BYPASS	R	1h	Bypass status signal. 1 CLKOUT in bypass

### 5.2.1.119 PLL\_PER\_HSDIVIDER Register (Offset = 864h) [Reset = 0000XXXXh]

PLL\_PER\_HSDIVIDER is shown in [Table 5-122](#).

Return to the [Summary Table](#).

**Table 5-122. PLL\_PER\_HSDIVIDER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	LDOPWDNACK	R	0h	LDO Power Down Ack
16	BYPASSACKZ	R	0h	HSDIVIDER Bypass Ack
15-3	RESERVED	R/W	0h	
2	TENABLEDIV	R/W	0h	Tenable Div
1	LDOPWDN	R/W	0h	LDO Power Down
0	BYPASS	R/W	0h	HSDIVIDER Bypass

### 5.2.1.120 PLL\_PER\_HSDIVIDER\_CLKOUT0 Register (Offset = 868h) [Reset = 00000XX4h]

PLL\_PER\_HSDIVIDER\_CLKOUT0 is shown in [Table 5-123](#).

Return to the [Summary Table](#).

**Table 5-123. PLL\_PER\_HSDIVIDER\_CLKOUT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M4 divider and hence CLKOUT0 output 0h (R/W) = CLKOUT0 divider active 1h (R/W) = CLKOUT0 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT0 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT0 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT0_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M4, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

### 5.2.1.121 PLL\_PER\_HSDIVIDER\_CLKOUT1 Register (Offset = 86Ch) [Reset = 0000XX4h]

PLL\_PER\_HSDIVIDER\_CLKOUT1 is shown in [Table 5-124](#).

Return to the [Summary Table](#).

**Table 5-124. PLL\_PER\_HSDIVIDER\_CLKOUT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M5 divider and hence CLKOUT1 output 0h (R/W) = CLKOUT1 divider active 1h (R/W) = CLKOUT1 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT1 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT1 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT1_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M5, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

### 5.2.1.122 PLL\_PER\_HSDIVIDER\_CLKOUT2 Register (Offset = 870h) [Reset = 00000XX4h]

PLL\_PER\_HSDIVIDER\_CLKOUT2 is shown in [Table 5-125](#).

Return to the [Summary Table](#).

**Table 5-125. PLL\_PER\_HSDIVIDER\_CLKOUT2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M6 divider and hence CLKOUT2 output 0h (R/W) = CLKOUT2 divider active 1h (R/W) = CLKOUT2 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT2 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT2 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT2_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M6, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved



### 5.2.1.123 PLL\_PER\_HSDIVIDER\_CLKOUT3 Register (Offset = 874h) [Reset = 00000XX4h]

PLL\_PER\_HSDIVIDER\_CLKOUT3 is shown in [Table 5-126](#).

Return to the [Summary Table](#).

**Table 5-126. PLL\_PER\_HSDIVIDER\_CLKOUT3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	PWDN	R/W	0h	Power down for HSDIVIDER M7 divider and hence CLKOUT3 output 0h (R/W) = CLKOUT3 divider active 1h (R/W) = CLKOUT3 divider is powered down
11-10	RESERVED	R/W	0h	
9	STATUS	R	0h	HSDIVIDER CLKOUT3 status 0h (R) = The clock output is gated 1h (R) = The clock output is enabled
8	GATE_CTRL	R/W	0h	Control gating of HSDIVIDER CLKOUT3 0h (R/W) = Automatically gate this clock when there is no dependency for it 1h (R/W) = Force this clock to stay enabled even if there is no request
7-6	RESERVED	R/W	0h	
5	DIVCHACK	R	0h	Toggle on this status bit after changing HSDIVIDER_CLKOUT3_DIV indicates that the change in divider value has taken effect
4-0	DIV	R/W	4h	DPLL post-divider factor, M7, for internal clock generation. Divide values from 1 to 31. 0h (R/W) = Reserved

### 5.2.1.124 PLL\_DSP\_RSTCTRL Register (Offset = 878h) [Reset = 00000000h]

PLL\_DSP\_RSTCTRL is shown in [Table 5-127](#).

Return to the [Summary Table](#).

**Table 5-127. PLL\_DSP\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the PLL Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.125 PLL\_DSP\_HSDIVIDER\_RSTCTRL Register (Offset = 87Ch) [Reset = 00000000h]

PLL\_DSP\_HSDIVIDER\_RSTCTRL is shown in [Table 5-128](#).

Return to the [Summary Table](#).

**Table 5-128. PLL\_DSP\_HSDIVIDER\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the HSDIVIDER Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.126 PLL\_PER\_RSTCTRL Register (Offset = 880h) [Reset = 00000000h]

PLL\_PER\_RSTCTRL is shown in [Table 5-129](#).

Return to the [Summary Table](#).

**Table 5-129. PLL\_PER\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the PLL Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.127 PLL\_PER\_HSDIVIDER\_RSTCTRL Register (Offset = 884h) [Reset = 00000000h]

PLL\_PER\_HSDIVIDER\_RSTCTRL is shown in [Table 5-130](#).

Return to the [Summary Table](#).

**Table 5-130. PLL\_PER\_HSDIVIDER\_RSTCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	SW Reset override for the HSDIVIDER Write 3'b111 : Override is enabled and Reset is asserted

### 5.2.1.128 ANA\_REG\_CLK\_CTRL\_REG1\_XO\_SLICER Register (Offset = C00h) [Reset = 00000000h]

ANA\_REG\_CLK\_CTRL\_REG1\_XO\_SLICER is shown in [Table 5-131](#).

Return to the [Summary Table](#).

**Table 5-131. ANA\_REG\_CLK\_CTRL\_REG1\_XO\_SLICER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
14	SLICER_APLL_BYPASS_DRV	R/W	0h	Slicer APLL Bypass Drive This bit controls the drive strength of the APLL Bypass Slicer 0 = Low-power drive 1 = High-power drive 0x0 = Functional Reset
13	SLICER_APLL_BYPASS	R/W	0h	Slicer APLL Bypass This bit enables a high-speed slicer connected to CLKM which can be used to drive a high-speed clock directly as the SYNTH reference clock. 0 = Normal operation (bypass slicer disabled) 1 = APLL Bypass Slicer Enabled 0x0 = Functional Reset
12	XTAL_DETECT_XO_SLICER	R/W	0h	XTAL Detect Enable This bit connects a pullup and sense circuitry to CLKM to detect the presence or absence of a crystal. This operation will conflict with oscillator functionality, so this bit must be asserted only when the oscillator is disabled (CTRL_CLKTOP_REG1 bit 2 must be '0'). After asserted, the internal XTAL_SENSE signal will reflect a '1' if a crystal is present (CLKM sees a high impedance) or '0' if CLKM is tied to ground. After the sense operation is detected, this bit must be cleared before the oscillator will function properly if enabled. 0 = Normal operation (pullup and sense circuitry are disconnected from CLKM, XTAL_SENSE outputs '1') 1 = XTAL sense function enabled (pullup and sense circuitry connected to CLKM, output of XTAL_SENSE reads '1' if high impedance, '0' if CLKM is tied to ground) 0x0 = Functional Reset
11	SLICER_DCCPL_XO_SLICER	R/W	0h	Slicer DC-Coupled Mode 0 = Normal operation (AC-couple CLKP to internal slicer) 1 = DC-couple CLKP to internal slicer to CLKP 0x0 = Functional Reset
10	SLICER_HIPWR_XO_SLICER	R/W	0h	Slicer High-power Mode This bit bypasses the input clock slicer current-starving/filtering circuitry to increase gain and reduce device phase-noise at the expense of power and reduced supply noise rejection. This permits the use of a high-speed external test clock (660MHz max). 0 = Normal operation (current-limiting present) 1 = High-power/high-speed test mode 0x0 = Functional Reset
9	FASTCHARGEZ_BIAS_XO_SLICER	R/W	0h	Bias Fast-charge Enable (Active Low) This bit bypasses the RC filtering on the XOSC/SLICER Bias to permit more rapid power-up. 0 = Bias fast-charge 1 = Normal operation (filtering present) 0x1 = Functional Reset
8-4	XOSC_DRIVE_XO_SLICER	R/W	0h	Crystal Oscillator Output Drive Binary-weighted oscillator drive control 0x0 = Functional Reset
3-0	RTRIM_BIAS_XO_SLICER	R/W	0h	Crystal Oscillator and Slicer Bias RTrim Binary-weighted bias control 0x0 = Functional Reset

### 5.2.1.129 ANA\_REG\_CLK\_CTRL\_REG1\_CLKTOP Register (Offset = C04h) [Reset = 00000000h]

ANA\_REG\_CLK\_CTRL\_REG1\_CLKTOP is shown in [Table 5-132](#).

Return to the [Summary Table](#).

**Table 5-132. ANA\_REG\_CLK\_CTRL\_REG1\_CLKTOP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
2	ENABLE_XOSC	R/W	0h	Enable Crystal Oscillator 0 = Disabled 1 = Enabled 0x1 = Functional Reset
1	ENABLE_SLICER_CLKP	R/W	0h	Enable CLKP Input Slicer 0 = Disabled 1 = Enabled 0x1 = Functional Reset
0	ENABLE_BIAS_XO_SLICER	R/W	0h	Enable Bias for Crystal Oscillator and Slicer 0 = Disabled 1 = Enabled 0x1 = Functional Reset

### 5.2.1.130 ANA\_REG\_CLK\_CTRL\_REG2\_CLKTOP Register (Offset = C08h) [Reset = 00000000h]

ANA\_REG\_CLK\_CTRL\_REG2\_CLKTOP is shown in [Table 5-133](#).

Return to the [Summary Table](#).

**Table 5-133. ANA\_REG\_CLK\_CTRL\_REG2\_CLKTOP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset



### 5.2.1.131 ANA\_REG\_CLK\_CTRL\_REG1\_LDO\_CLKTOP Register (Offset = C0Ch) [Reset = 00000000h]

ANA\_REG\_CLK\_CTRL\_REG1\_LDO\_CLKTOP is shown in [Table 5-134](#).

Return to the [Summary Table](#).

**Table 5-134. ANA\_REG\_CLK\_CTRL\_REG1\_LDO\_CLKTOP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
9	CLK_BIST_DISABLE_LDO	R/W	0h	DC BIST Disable for LDO 0 = Normal operation of DC BIST 1 = DC BIST Disabled 0x0 = Functional Reset
8-1	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
0	EN_SLICER_LDO	R/W	0h	SLICER LDO ENABLE 0 = Slicer LDO Disabled 1 = Slicer LDO Enabled 0x1 = Functional Reset

### 5.2.1.132 ANA\_REG\_CLK\_CTRL\_REG2\_LDO\_CLKTOP Register (Offset = C10h) [Reset = 00000000h]

ANA\_REG\_CLK\_CTRL\_REG2\_LDO\_CLKTOP is shown in [Table 5-135](#).

Return to the [Summary Table](#).

**Table 5-135. ANA\_REG\_CLK\_CTRL\_REG2\_LDO\_CLKTOP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
23-20	BISTMUX_CTRL	R/W	0h	SLICER LDO BIST MUX CONTROL (ONE HOT) Analog MUX enables to BIST output port 0000 = HI-Z Output 0001 = VBG_0P9*10/9 = 1.0 V 0010 = VDD18*0.5 = 0.9V 0100 = VLDO Output * 0.6 1000 = Floating WARNING: Enabling more than one bit may damage the device 0x4 = Functional Reset
19-16	TESTMUX_CTRL	R/W	0h	SLICER LDO TEST MUX CONTROL (ONE HOT) Analog MUX enables to test output port 0000 = HI-Z Output 0001 = 0.6 * VLDO_OUT 0010 = VDD18*0.5 = 0.9V 0100 = VSSA 1000 = LDO Test Current (12.5uA) WARNING: Enabling more than one bit may damage the device 0x0 = Functional Reset
15-13	TLOAD_CTRL	R/W	0h	SLICER LDO TLOAD CONTROL Need inverter on bit13 updated description needed 0x0 = Functional Reset
12	ENABLE_Pmos_PULLDOWN	R/W	0h	SLICER LDO PMOS PULL DOWN ENABLE 0 = Slicer LDO PMOS Pull Down disabled 1 = Slicer LDO PMOS Pull Down enabled 0x0 = Functional Reset
11	SCPRT_IBIAS_CTRL	R/W	0h	SLICER LDO SHORT CKT PROTECTION IBIAS CONTROL 0 = Nominal short circuit bias with nominal short circuit current limit 1 = 2X Nominal short circuit bias with higher short circuit current limit 0x0 = Functional Reset
10-8	LDO_BW_CTRL	R/W	0h	SLICER LDO BANDWIDTH CONTROL Need inverters on bits 8 and 10 need updated description 0x7 = Functional Reset
7	EN_BYPASS	R/W	0h	SLICER LDO BYPASS ENABLE 0 = Slicer LDO in normal mode 1 = Slicer LDO Bypassed with external voltage 0x0 = Functional Reset
6	EN_SHRT_CKT	R/W	0h	SLICER LDO SHORT CKT PROTECTION ENABLE 0 = Slicer LDO Short Ckt Protection Disabled 1 = Slicer LDO Short Ckt Protection Enabled 0x0 = Functional Reset
5	EN_TEST_MODE	R/W	0h	SLICER LDO TEST MODE ENABLE 0 = Slicer LDO TEST MODE Disabled 1 = Slicer LDO TEST MODE Enabled 0x0 = Functional Reset
4	ENZ_LOW_BW_CAP	R/W	0h	SLICER LDO LOW BW MODE DISABLE 1 = Slicer LDO Low BW mode Enabled 0 = Slicer LDO Low BW mode Disabled Description IS updated above Need inverter 0x1 = Functional Reset
3-0	LDO_VOUT_CTRL	R/W	0h	SLICER LDO VOUT TRIM NEEDS updated description Need inverters on 0,1,2 0x0 = Functional Reset

### 5.2.1.133 ANA\_REG\_CLK\_STATUS\_REG Register (Offset = C18h) [Reset = 00000000h]

ANA\_REG\_CLK\_STATUS\_REG is shown in [Table 5-136](#).

Return to the [Summary Table](#).

**Table 5-136. ANA\_REG\_CLK\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	Reserved Reads return 0x0 and writes have no effect.
8	CLK_TEST_PATH_LDO_SC_OUT	R	0h	CLK TEST PATH LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
7	SYNC_20G_LDO_SC_OUT	R	0h	SYNC 20G LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
6	CLKTOP_IOBUF_ROUTE_LDO_SC_OUT	R	0h	CLKTOP IOBUF ROUTE LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
5	SYNTH_DIV_LDO_SC_OUT	R	0h	SYNTH DIV LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
4	SYNTH_VCO_LDO_SC_OUT	R	0h	SYNTH VCO LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
3	SDM_LDO_SC_OUT	R	0h	SDM LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
2	CLKTOP_IOBUF_APLL_LDO_SC_OUT	R	0h	CLKTOP IOBUF APLL LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
1	APLL_VCO_LDO_SC_OUT	R	0h	APLL VCO LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected
0	SLICER_LDO_SC_OUT	R	0h	SLICER LDO SHORT CIRCUIT INDICATOR 0 = Normal operation 1 = LDO Output Short Circuit Detected

### 5.2.1.134 ANA\_REG\_REFSYS\_CTRL\_REG\_LOWV Register (Offset = C1Ch) [Reset = 00000000h]

ANA\_REG\_REFSYS\_CTRL\_REG\_LOWV is shown in [Table 5-137](#).

Return to the [Summary Table](#).

**Table 5-137. ANA\_REG\_REFSYS\_CTRL\_REG\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved 0x0 = Functional Reset
30-27	FTRIM_3_0	R/W	0h	Filter TRIM Control 0x0 = Functional Reset
26	DO_NOT_USE3	R/W	0h	Do not use this bit --> mapped to REFSYS_CTRL_REG<26> 0x0 = Functional Reset
25	DO_NOT_USE2	R/W	0h	Do not use this bit --> mapped to REFSYS_CTRL_REG<25> 0x0 = Functional Reset
24	REFSYS_BYPASS_EN	R/W	0h	<5> REFSYS By-Pass Enable 0x0 = Functional Reset
23	DO_NOT_USE1	R/W	0h	Do not use this bit --> mapped to REFSYS_CTRL_REG<23> 0x0 = Functional Reset
22	DO_NOT_USE0	R/W	0h	Do not use this bit --> mapped to REFSYS_CTRL_REG<22> 0x0 = Functional Reset
21	CLKTOP_IBIAS_EN	R/W	0h	<2> CLK TOP IBIAS EN 0x1 = Functional Reset
20	V2I_STARTUP	R/W	0h	<1> V2I Startup 0x0 = Functional Reset
19	BGAP_ISW	R/W	0h	<0> BGAP ISW STARTUP 0x0 = Functional Reset
18-14	IREF_TRIM_4_0	R/W	0h	Default Resistor Trim for NOM LOT 0x02 = Functional Reset
13-9	MAG_TRIM_4_0	R/W	0h	Default Magnitude Trim for NOM LOT 0x00 = Functional Reset
8-4	SLOPE_TRIM_4_0	R/W	0h	Default Slope Trim for NOM LOT 0x0D = Functional Reset
3	REFSYS_PRE_CHARGE	R/W	0h	REFSYS Pre Charge Control 0 = Disable Pre Charge Block 1 = Enable Pre Charge Block 0x0 = Functional Reset
2	REFSYS_CAP_SW_CTRL LZ	R/W	0h	REFSYS Cap Switch Control 0 = Switch External Cap to reference output 1 = Disconnect External Cap to Reference output 0x0 = Functional Reset
1	REFSYS_V2I_EN_CTRL	R/W	0h	REFSYS Enable Control 0 = Disable V2I REFSYS 1 = Enable V2I REFSYS 0x1 = Functional Reset
0	REFSYS_BGAP_EN_CTRL L	R/W	0h	REFSYS Enable Control 0 = Disable REFSYS 1 = Enable REFSYS 0x1 = Functional Reset

### 5.2.1.135 ANA\_REG\_REFSYS\_TMUX\_CTRL\_LOWV Register (Offset = C20h) [Reset = 00000000h]

ANA\_REG\_REFSYS\_TMUX\_CTRL\_LOWV is shown in [Table 5-138](#).

Return to the [Summary Table](#).

**Table 5-138. ANA\_REG\_REFSYS\_TMUX\_CTRL\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	REFSYS_CTRL_8	R/W	0h	REFSYS Test Mux Enable. Other bits in Bus are One-hot. This control enabled in sync with other one hot control bits in Reg 0 = TMUX Disabled 1 = TMUX Enabled 0x0 = Functional Reset
30-16	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0000 = Functional Reset
15	LO_IBIASP_20u	R/W	0h	<15> LO IBG BIASP 20uA (TMUX One-Hot) 0x0 = Functional Reset
14	TX_IBIASP_20u	R/W	0h	<14> TX IBG BIASP 20uA (TMUX One-Hot) 0x0 = Functional Reset
13	BYPASS_MIRR_VPBIAS	R/W	0h	VPBIAS Control for IREF Gen Test Mode V2I By-Pass Feature 0x0 = Functional Reset
12	I2V_SENSE	R/W	0h	Sense Voltage from the BIST I2V conversion of 20u and 12u bias current paths Sense voltage of 1V for BIST select<6> Sense voltage of 0.6V for BIST select<7> 0x0 = Functional Reset
11	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
10	IREFP_10UA	R/W	0h	<10> IREFP 10uA (TMUX One-Hot) 0x0 = Functional Reset
9	IDIODEP_100U	R/W	0h	<9> Idiode BIASP 100uA (TMUX One-Hot) 0x0 = Functional Reset
8	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
7	IBIASP_TS_6U	R/W	0h	<7> IBG BIASP TS 6uA (TMUX One-Hot) 0x0 = Functional Reset
6	IBIASP_20U	R/W	0h	<6> CLK IBG BIASP 20uA (TMUX One-Hot) 0x0 = Functional Reset
5	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
4	VBE_WEAK	R/W	0h	<4> - VBE Weak (TMUX One-Hot) 0x0 = Functional Reset
3	RX_IBIASP_20u	R/W	0h	<3> RX IBG BIASP 20uA (TMUX One-Hot) 0x0 = Functional Reset
2	VBG_1P22V	R/W	0h	<2> - VBG 1.22V (TMUX One-Hot) 0x0 = Functional Reset
1	VREF_0P9V	R/W	0h	<1> - VREF 0P9V (Cap Node) (TMUX One-Hot) 0x0 = Functional Reset
0	VREF_0P45V	R/W	0h	<0> - VREF 0P45 (TMUX One-Hot) 0x0 = Functional Reset

### 5.2.1.136 ANA\_REG\_REFSYS\_SPARE\_REG\_LOWV Register (Offset = C24h) [Reset = 00000000h]

ANA\_REG\_REFSYS\_SPARE\_REG\_LOWV is shown in [Table 5-139](#).

Return to the [Summary Table](#).

**Table 5-139. ANA\_REG\_REFSYS\_SPARE\_REG\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved Used for ANALOGTEST TMUX ESD CTRL in Pad-Frame in TPR (formerly RX_REFSYS_TMUX_SPARE_CTRL_LOWV<31> in AWR/IWR devices, but RX does not exist in TPR) 0x0 = Functional Reset
30-28	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
27-26	VDDA_OSC_IR_DROP_C OMP_SEL	R/W	0h	VDDA_OSC UV VMON Reference Selection 0x0 = 0.56V 0x1 = 0.54V 0x2 = 0.52V 0x3 = 0.5V 0x0 = Functional Reset
25-24	VDDS_3P3V_IR_DROP_ COMP_SEL	R/W	0h	VIOIN VMON UV Reference Selection 0x0 = 0.56V 0x1 = 0.54V 0x2 = 0.52V 0x3 = 0.5V 0x0 = Functional Reset
23-22	VDD_IR_DROP_COMP_S EL	R/W	0h	VDD 1.2V VMON UV Reference Selection and VDD 1.2V VMON OV Self-test Reference Selection If MSS_REFSYS_SPARE_REG<8> = 0x0, reference selection is dependent on MSS_REFSYS_SPARE_REG<7:6> programming (normal VDD 1.2V VMON UV operation) If MSS_REFSYS_SPARE_REG<9> = 0x1, reference selection is dependent on MSS_REFSYS_SPARE_REG<17:16> programming (VDD 1.2V VMON OV Self-test operation) NOTE: MSS_REFSYS_SPARE_REG<9> != MSS_REFSYS_SPARE_REG<8> is invalid Reference selection is dependent on MSS_REFSYS_SPARE_REG<7:6> programming If MSS_REFSYS_SPARE_REG<7:6> = 0x0 (or MSS_REFSYS_SPARE_REG<17:16> = 0x0) 0x0 = 0.58V 0x1 = 0.57V 0x2 = 0.56V 0x3 = 0.55V If MSS_REFSYS_SPARE_REG<7:6> = 0x1 (or MSS_REFSYS_SPARE_REG<17:16> = 0x1) 0x0 = 0.55V 0x1 = 0.54V 0x2 = 0.53V 0x3 = 0.52V If MSS_REFSYS_SPARE_REG<7:6> = 0x2 (or MSS_REFSYS_SPARE_REG<17:16> = 0x2) 0x0 = 0.53V 0x1 = 0.52V 0x2 = 0.51V 0x3 = 0.5V If MSS_REFSYS_SPARE_REG<7:6> = 0x3 (or MSS_REFSYS_SPARE_REG<17:16> = 0x3) 0x0 = 0.51V 0x1 = 0.5V 0x2 = 0.49V 0x3 = 0.48V 0x0 = Functional Reset
21	VDD_OV_RSET_MASK	R/W	0h	If asserted, VDD_OV will not trigger the automatic reset of the device through WU Seq hardware control. However, OV flag will still propagate to the digital where the CPU will need to take action. 0x0 = Functional Reset
20	VDD_UV_RSET_MASK	R/W	0h	If asserted, VDD_UV will not trigger the automatic reset of the device through WU Seq hardware control. However, UV flag will still propagate to the digital where the CPU will need to take action. 0x0 = Functional Reset
19	VDDA_OSC_UV_RSET_ MASK	R/W	0h	If asserted, VDDA_OSC_UV will not trigger the automatic reset of the device through WU Seq hardware control. However, UV flag will still propagate to the digital where the CPU will need to take action. 0x0 = Functional Reset
18	VIOIN_UV_RSET_MASK	R/W	0h	If asserted, VIOIN_UV will not trigger the automatic reset of the device through WU Seq hardware control. However, UV flag will still propagate to the digital where the CPU will need to take action. 0x0 = Functional Reset
17-16	VDD_OV_SR_SEL	R/W	0h	Final level of VDD 1.2V VMON OV Reference Selection See definition in MSS_REFSYS_SPARE_REG<15:14> for normal operation (MSS_REFSYS_SPARE_REG<9> = 0x0) See definition in MSS_REFSYS_SPARE_REG<23:22> for self-test operation (MSS_REFSYS_SPARE_REG<9> = 0x1) 0x0 = Functional Reset

**Table 5-139. ANA\_REG\_REFSYS\_SPARE\_REG\_LOWV Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
15-14	VDD_OV_IR_DROP_CO MP_SEL	R/W	0h	VDD 1.2V VMON OV Reference Selection and VDD 1.2V VMON UV Self-test Reference Selection If MSS_REFSYS_SPARE_REG<9> = 0x0, reference selection is dependent on MSS_REFSYS_SPARE_REG<17:16> programming (normal VDD 1.2V VMON OV operation) If MSS_REFSYS_SPARE_REG<8> = 0x1, reference selection is dependent on MSS_REFSYS_SPARE_REG<7:6> programming (VDD 1.2V VMON UV Self-test operation) NOTE: MSS_REFSYS_SPARE_REG<9> ! = MSS_REFSYS_SPARE_REG<8> is invalid If MSS_REFSYS_SPARE_REG<17:16> = 0x0 (or MSS_REFSYS_SPARE_REG<7:6> = 0x0) 0x0 = 0.68V 0x1 = 0.67V 0x2 = 0.66V 0x3 = 0.65V If MSS_REFSYS_SPARE_REG<17:16> = 0x1 (or MSS_REFSYS_SPARE_REG<7:6> = 0x1) 0x0 = 0.65V 0x1 = 0.64V 0x2 = 0.63V 0x3 = 0.62V If MSS_REFSYS_SPARE_REG<17:16> = 0x2 (or MSS_REFSYS_SPARE_REG<7:6> = 0x2) 0x0 = 0.62V 0x1 = 0.61V 0x2 = 0.6V 0x3 = 0.59V If MSS_REFSYS_SPARE_REG<17:16> = 0x3 (or MSS_REFSYS_SPARE_REG<7:6> = 0x3) 0x0 = 0.59V 0x1 = 0.58V 0x2 = 0.57V 0x3 = 0.56V 0x0 = Functional Reset
13	RESERVED	R/W	0h	Reserved Reserved in case VIOIN OV VMON and self test is ever implemented 0x0 = Functional Reset
12	VDDS_3P3V_UV_SELF_ TEST_SEL	R/W	0h	Enable VIOIN Strict UV VMON Self Test If Self-test mode is enabled, VIOIN UV VMON reference is programmed as follows for MSS_REFSYS_SPARE_REG<25:24>: 0x0 = 0.66V 0x1 = 0.64V 0x2 = 0.62V 0x3 = 0.6V 0x0 = Functional Reset
11	RESERVED	R/W	0h	Reserved Reserved in case VDDA_OSC OV VMON and self test is ever implemented 0x0 = Functional Reset
10	VDDA_OSC_UV_SELF_T EST_SEL	R/W	0h	Enable VDDA_OSC Strict UV VMON Self Test If Self-test mode is enabled, VDDA_OSC UV VMON reference is programmed as follows for MSS_REFSYS_SPARE_REG<27:26>: 0x0 = 0.66V 0x1 = 0.64V 0x2 = 0.62V 0x3 = 0.6V 0x0 = Functional Reset
9	VDD_OV_SELF_TEST_S EL	R/W	0h	Enable 1.2V VDD Strict OV VMON Self Test If Self-test mode is enabled, VDD 1.2V VMON OV reference is programmed based on MSS_REFSYS_SPARE_REG<23:22> and MSS_REFSYS_SPARE_REG<17:16> as follows: If MSS_REFSYS_SPARE_REG<17:16> = 0x0, MSS_REFSYS_SPARE_REG<23:22>: 0x0 = 0.58V 0x1 = 0.57V 0x2 = 0.56V 0x3 = 0.55V If MSS_REFSYS_SPARE_REG<17:16> = 0x1, MSS_REFSYS_SPARE_REG<23:22>: 0x0 = 0.55V 0x1 = 0.54V 0x2 = 0.53V 0x3 = 0.52V If MSS_REFSYS_SPARE_REG<17:16> = 0x2, MSS_REFSYS_SPARE_REG<23:22>: 0x0 = 0.53V 0x1 = 0.52V 0x2 = 0.51V 0x3 = 0.5V If MSS_REFSYS_SPARE_REG<17:16> = 0x3, MSS_REFSYS_SPARE_REG<23:22>: 0x0 = 0.51V 0x1 = 0.5V 0x2 = 0.49V 0x3 = 0.48V 0x0 = Functional Reset
8	VDD_UV_SELF_TEST_S EL	R/W	0h	Enable 1.2V VDD Strict UV VMON Self Test If Self-test mode is enabled, VDD 1.2V VMON UV reference is programmed based on MSS_REFSYS_SPARE_REG<15:14> and MSS_REFSYS_SPARE_REG<7:6> as follows: If MSS_REFSYS_SPARE_REG<7:6> = 0x0, MSS_REFSYS_SPARE_REG<15:14>: 0x0 = 0.68V 0x1 = 0.67V 0x2 = 0.66V 0x3 = 0.65V If MSS_REFSYS_SPARE_REG<7:6> = 0x1, MSS_REFSYS_SPARE_REG<15:14>: 0x0 = 0.65V 0x1 = 0.64V 0x2 = 0.63V 0x3 = 0.62V If MSS_REFSYS_SPARE_REG<7:6> = 0x2, MSS_REFSYS_SPARE_REG<15:14>: 0x0 = 0.62V 0x1 = 0.61V 0x2 = 0.6V 0x3 = 0.59V If MSS_REFSYS_SPARE_REG<7:6> = 0x3, MSS_REFSYS_SPARE_REG<15:14>: 0x0 = 0.59V 0x1 = 0.58V 0x2 = 0.57V 0x3 = 0.56V 0x0 = Functional Reset
7-6	VDD_SR_SEL	R/W	0h	Final level of VDD 1.2V VMON UV Reference Selection See definition in MSS_REFSYS_SPARE_REG<23:22> for normal operation (MSS_REFSYS_SPARE_REG<8> = 0x0) See definition in MSS_REFSYS_SPARE_REG<15:14> for self-test operation (MSS_REFSYS_SPARE_REG<8> = 0x1) 0x0 = Functional Reset

**Table 5-139. ANA\_REG\_REFSYS\_SPARE\_REG\_LOWV Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5-1	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect, but these bits are tied to efuse overrides. 0x0 = Functional Reset
0	RESERVED	R/W	0h	Reserved In TPR, this bit is mapped to efuse and is used to control POR_DIG_SEQ_ECO_DIS. Writes have no effect. 0: Enable Slicer delay ECO 1: Disable Slicer delay ECO 0x0 = Functional Reset



### 5.2.1.137 ANA\_REG\_WU\_CTRL\_REG\_LOWV Register (Offset = C28h) [Reset = 00000000h]

ANA\_REG\_WU\_CTRL\_REG\_LOWV is shown in [Table 5-140](#).

Return to the [Summary Table](#).

**Table 5-140. ANA\_REG\_WU\_CTRL\_REG\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x0 = Functional Reset
30-29	WU_SPARE_IN_2	R/W	0h	WU Spare Control 0x3 = Functional Reset
28	WU_VDD_OV_VMON_EN	R/W	0h	WU VDD OV VMON Enable Control 0 = VDD OV Detect Disabled 1 = VDD OV Detect Enabled 0x0 = Functional Reset
27	WU_VDD_UV_VMON_EN	R/W	0h	WU VDD UV VMON Enable Control 0 = VDD UV Detect Disabled 1 = VDD UV Detect Enabled 0x0 = Functional Reset
26	WU_VDDA_OSC_UV_VMON_EN	R/W	0h	WU VDDA OSC UV VMON Enable Control 0 = VDDA OSC UV Detect Disabled 1 = VDDA OSC UV Detect Enabled 0x0 = Functional Reset
25	WU_VDDS_3P3V_UV_VMON_EN	R/W	0h	WU VDDS 3.3V UV VMON Enable Control 0 = VDDS 3.3V UV Detect Disabled 1 = VDDS 3.3V UV Detect Enabled 0x0 = Functional Reset
24-23	WU_SPARE_IN	R/W	0h	WU Spare Control Change for 1642 ES2P0 Change Name : Newly added OR gates to provide options to bypass crude VDD DET (also refer to <11>) Bit <0> of this field when HIGH over rides the crude VDD_DET, this control is using firmware Bit<0> of this field is WU_CTRL_REG<23> Change for 2243 ES1P0 Using Bit<1>, is WU_CTRL_REG<24> This bit is used to make the reset fix SW controllable. Manshul's email notes-- Since the default value of WU_SPARE_IN<1> is '0', we will have the fix active by default. To disable it, firmware can write this bit to '1' before enabling VMON. 0x0 = Functional Reset
22	WU_SUPP_DET_CTRL	R/W	0h	WU VMON Detect Status Override Disable in Functional Test SOP 0 = VMON Det Status Override Disabled 1 = VMON Det Status Override Enabled 0x1 = Functional Reset
21	WU_VRAM_VMON_EN	R/W	0h	WU VRAM VMON Enable Control 0 = SRAM UV Detect Disabled 1 = SRAM UV Detect Enabled 0x1 = Functional Reset
20	WU_SUPP_VMON_EN	R/W	0h	WU VMON Enable Control 0 = VMON Control Disabled 1 = VMON Control Enabled 0x1 = Functional Reset
19	WU_XTAL_DLY_CTRL	R/W	0h	Introduce additional delay for XTAL settling 0 = Default delay as per WU-SEQ 1 = Introduce additional delay as per WU-SEQ 0x0 = Functional Reset
18	WU_OV_DET_CTRL	R/W	0h	WU Over Voltage Detect Control Changed for 1243 ES3P0 (Metal only change from 1642 ES2P0) Change Name : FW control of VDD OV DET EN 1 = OV Detect is disabled 0 = OV Detect is Enabled 0x1 = Functional Reset
17	WU_UV_DET_CTRL	R/W	0h	WU Under Voltage Detect Control 0 = UV Detect is disabled 1 = UV Detect is Enabled 0x1 = Functional Reset
16	XTAL_EN_OVERRIDE	R/W	0h	XTAL EN Override (WU-SEQ) Control 0 = XTAL Enable is driven by WU-SEQ detection 1 = Override XTAL Enable if disabled by default 0x0 = Functional Reset
15	WU_CPU_CLK_CTRL	R/W	0h	WU CLK Control 0 = CLK Monitor Function in Dig Sequencer is disabled 1 = REF CLK Monitor Function is Enabled 0x1 = Functional Reset

**Table 5-140. ANA\_REG\_WU\_CTRL\_REG\_LOWV Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14-11	INT_CLK_FREQ_SEL_3_0	R/W	0h	WU Internal Clock (RCOSC) Frequency Select Bit<3> is used as override for VMON on Untrimmed devices. Bit <3> is '1' if device REFSYS_TOP is trimmed. Changed on 1642 ES2P0 Change Name : Newly added mux for CLK MON EN options When Bit<2> = 0, MASK_CPU_CLK_OUT_CTRL_LOWV == WU_CTRL_REG<12>, essentially Bit<1> of this field When Bit<2> = 1, MASK_CPU_CLK_OUT_CTRL_LOWV == (original function) INTER_MASK_CPU_CLK_OUT_CTRL_LOWV Change Name : Newly added OR gates to provide options to bypass crude VDD DET (also refer to <23>) Bit <0> of this field when HIGH overrides the crude VDD_DET 0x0 = Functional Reset
10	ECO_SLICER_CLK_DLY_DIS	R/W	0h	Disable slicer clock delay ECO (mapped to eFuse) 0x0 = Functional Reset
9-3	INT_CLK_TRIM_6_0	R/W	0h	WU Internal Clock (RCOSC) Trim 0x4B = Functional Reset (If not trimmed)
2	INT_CLK_SW_SEL	R/W	0h	WU Internal Clock (RCOSC) SW_SEL 0 = TBD 1 = TBD 0x1 = Functional Reset
1	INT_CLK_STOP	R/W	0h	WU Internal Clock (RCOSC) STOP 0 = Internal CLK can be enabled 1 = Internal CLK is OFF 0x0 = Functional Reset
0	INT_CLK_EN	R/W	0h	WU Internal Clock (RCOSC) ENABLE 0 = Internal CLK Disabled 1 = Internal CLK Enabled 0x1 = Functional Reset

### 5.2.1.138 ANA\_REG\_WU\_TMUX\_CTRL\_LOWV Register (Offset = C2Ch) [Reset = 00000000h]

ANA\_REG\_WU\_TMUX\_CTRL\_LOWV is shown in [Table 5-141](#).

Return to the [Summary Table](#).

**Table 5-141. ANA\_REG\_WU\_TMUX\_CTRL\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	WU_TMUX_EN	R/W	0h	WU TMUX Enable 0 = TMUX Disabled 1 = TMUX Enabled 0x0 = Functional Reset
30-25	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x00 = Functional Reset
24	SCALED_VDDS33	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $0.289 * VDDS33$ 0x0 = Functional Reset
23	SCALED_VDDA_LVDS_1 P8V	R/W	0h	Test Mux Control. One Hot Control Change in TPR VDDS_3P3V IO DET reference (3.3V mode) Scaling Factor: $0.4 * VDDA\_LVDS\_1P8V$ 0x0 = Functional Reset
22	SCALED_VDDA_LVDS_1 P8V_1P2	R/W	0h	Test Mux Control. One Hot Control Change in TPR VDDS_3P3V IO DET reference (1.8V mode) Scaling Factor: $0.67 * VDDA\_LVDS\_1P8V$ 0x0 = Functional Reset
21	VFB_0P85V	R/W	0h	Scaled VDD 1.2V used as reference for VDDA_OSC crude supply detect 0x0 = Functional Reset
20	VDDSINT18	R/W	0h	VIOIN scaled supply for VIOIN Detect Scaling Factor: $VIOIN * (52/90)$ 0x0 = Functional Reset
19	SCALED_VDDA_OSC_DI V22_39	R/W	0h	Scaled VDDA_OSC supply for crude supply detect Scaling Factor: $VDDA\_OSC * (22/39)$ 0x0 = Functional Reset
18	VT_DIG_SIG_OV	R/W	0h	Test Mux Control. One Hot Control VT Detect Signal Level on VDD12 Crude OV VMON 0x0 = Functional Reset
17	VT_DIG_SIG_UV	R/W	0h	Test Mux Control. One Hot Control VT Detect Signal Level on VDD12 Crude UV VMON 0x0 = Functional Reset
16	VT_ANA_SIG	R/W	0h	Test Mux Control. One Hot Control VT Detect Signal Level on VDDA_LVDS_1P8V 0x0 = Functional Reset
15	VDDA10RF2_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDDA10RF2_UV VMON 0x0 = Functional Reset
14	VDDA10RF1_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDDA10RF1_UV VMON 0x0 = Functional Reset
13	VDDA18VCO_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDDA18VCO_UV VMON 0x0 = Functional Reset
12	VDD_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDD_UV VMON 0x0 = Functional Reset
11	VDD_OV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDD_OV VMON 0x0 = Functional Reset
10	VDDA18BB_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDDA18BB_UV VMON 0x0 = Functional Reset
9	VIOIN33_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VIOIN33_UV VMON 0x0 = Functional Reset
8	VDDA_OSC_UV_VREF	R/W	0h	Test Mux Control. One Hot Control Vref for VDDA_OSC_UV VMON 0x0 = Functional Reset
7	VREF_0P9V	R/W	0h	Test Mux Control. One Hot Control VREF_0P9V 0x0 = Functional Reset
6	SCALED_VDDA10RF2	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $VDDA10RF2/2$ 0x0 = Functional Reset
5	SCALED_VDDA10RF1	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $VDDA10RF1/2$ 0x0 = Functional Reset
4	SCALED_VDDA18VCO	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $VDDA18VCO/3$ 0x0 = Functional Reset

**Table 5-141. ANA\_REG\_WU\_TMUX\_CTRL\_LOWV Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
3	VFB_0P6V	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $0.5 * VDD$ 0x0 = Functional Reset
2	SCALED_VDDA18BB	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $VDDA18BB/3$ 0x0 = Functional Reset
1	SCALED_VIO3318	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $VIOIN/5.5$ (3.3V mode), $VIOIN/3$ (1.8V mode) 0x0 = Functional Reset
0	SCALED_VDDA_OSC_DI V3	R/W	0h	Test Mux Control. One Hot Control Scaling Factor: $VDDA\_OSC/3$ 0x0 = Functional Reset

### 5.2.1.139 ANA\_REG\_TW\_CTRL\_REG\_LOWV Register (Offset = C30h) [Reset = 00000000h]

ANA\_REG\_TW\_CTRL\_REG\_LOWV is shown in [Table 5-142](#).

Return to the [Summary Table](#).

**Table 5-142. ANA\_REG\_TW\_CTRL\_REG\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R/W	0h	Reserved 0x00000 = Functional Reset
14	ANA_TMUX_BUF_EN	R/W	0h	MSS TW ANA TMUX Buffer Enabled 0 = ANA TMUX Buffer Disabled 1 = ANA TMUX Buffer Enabled 0x0 = Functional Reset
13	ANA_TMUX_BUF_BYPASS	R/W	0h	MSS TW ANA TMUX Buffer Bypass 0 = ANA TMUX Buffer By-pass Disabled 1 = ANA TMUX Buffer By-pass Enabled 0x0 = Functional Reset
12	VIN_EXT_CTRL	R/W	0h	MSS TW VIN Control from External Source 0 = External VIN Control Disabled 1 = External VIN Control Enabled 0x0 = Functional Reset
11	VREF_EXT_CTRL	R/W	0h	MSS TW VREF Control from External Source 0 = External VREF Control Disabled 1 = External VREF Control Enabled 0x0 = Functional Reset
10	IFORCE_EXT_CTRL	R/W	0h	MSS TW Iforce Control from External Source 0 = IFORCE Control Disabled 1 = IFORCE Control Enabled 0x0 = Functional Reset
9	TS_SE_INP_BUF_EN	R/W	0h	MSS TW ADC TS SE Inp Buffer Enable 0 = Input Buffer disabled 1 = Input Buffer Enabled 0x0 = Functional Reset
8	TS_DIFF_INP_BUF_EN	R/W	0h	MSS TW ADC TS DIFF Inp Buffer Enable 0 = Input Buffer disabled 1 = Input Buffer Enabled 0x1 = Functional Reset
7-5	ADC_REF_SEL_2_0	R/W	0h	MSS TW ADC Reference Select 0b001 = Reference from Top Refsys 0b010 = Reference from RX Refsys 0b100 = Reference from External Test Pin (CZ/ Trim) 0x001 = Functional Reset
4	ADC_REF_BUF_EN	R/W	0h	MSS TW ADC Reference Buffer Enable 0 = Input Buffer disabled 1 = Input Buffer Enabled (Default) 0x1 = Functional Reset
3	ADC_INP_BUF_EN	R/W	0h	MSS TW ADC Input Buffer Enable 0 = Input Buffer disabled 1 = Input Buffer Enabled (Default) 0x1 = Functional Reset
2	ADC_RESET	R/W	0h	MSS TW ADC Reset (Active High) 0 = ADC Out of Reset 1 = ADC In Reset 0x1 = Functional Reset
1	RESERVED	R/W	0h	RESERVED the bit to analog is directly driven MSS_GPADC
0	ADC_EN	R/W	0h	TW MSS ADC Control 0 = ADC Disable 1 = ADC Enable 0x0 = Functional Reset

### 5.2.1.140 ANA\_REG\_TW\_ANA\_TMUX\_CTRL\_LOWV Register (Offset = C34h) [Reset = 00000000h]

ANA\_REG\_TW\_ANA\_TMUX\_CTRL\_LOWV is shown in [Table 5-143](#).

Return to the [Summary Table](#).

**Table 5-143. ANA\_REG\_TW\_ANA\_TMUX\_CTRL\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	DO_NOT_USE21	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<31> 0x0 = Functional Reset
30	DO_NOT_USE20	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<30> 0x0 = Functional Reset
29-23	RESERVED	R/W	0h	Reserved Reads return 0x0 and writes have no effect. 0x00 = Functional Reset
22	DO_NOT_USE19	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<22> 0x0 = Functional Reset
21	ADC_REF_BUF_OUT	R/W	0h	MSS ADC reference buffer out to Test Pin 0x0 = Functional Reset
20	ADC_BUF_OUT_1P8V	R/W	0h	Buffered output of MSS ADC inputs to GPADC 0x0 = Functional Reset
19	BIST_MUX_OUT_1P8V	R/W	0h	MSS BIST Mux output pre ADC input Buffer 0x0 = Functional Reset
18	DO_NOT_USE18	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<18> 0x0 = Functional Reset
17	DO_NOT_USE17	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<17> 0x0 = Functional Reset
16	DO_NOT_USE16	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<16> 0x0 = Functional Reset
15	DO_NOT_USE15	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<15> 0x0 = Functional Reset
14	DO_NOT_USE14	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<14> 0x0 = Functional Reset
13	DO_NOT_USE13	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<13> 0x0 = Functional Reset
12	DO_NOT_USE12	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<12> 0x0 = Functional Reset
11	DO_NOT_USE11	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<11> 0x0 = Functional Reset
10	DO_NOT_USE10	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<10> 0x0 = Functional Reset
9	DO_NOT_USE9	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<9> 0x0 = Functional Reset
8	DO_NOT_USE8	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<8> 0x0 = Functional Reset
7	DO_NOT_USE7	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<7> 0x0 = Functional Reset
6	DO_NOT_USE6	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<6> 0x0 = Functional Reset
5	DO_NOT_USE5	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<5> 0x0 = Functional Reset
4	DO_NOT_USE4	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<4> 0x0 = Functional Reset
3	DO_NOT_USE3	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<3> 0x0 = Functional Reset
2	DO_NOT_USE2	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<2> 0x0 = Functional Reset
1	DO_NOT_USE1	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<1> 0x0 = Functional Reset

**Table 5-143. ANA\_REG\_TW\_ANA\_TMUX\_CTRL\_LOWV Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	DO_NOT_USE0	R/W	0h	Do not use this bit --> mapped to TW_ANA_TMUX_CTRL<0> 0x0 = Functional Reset

### 5.2.1.141 ANA\_REG\_WU\_MODE\_REG\_LOWV Register (Offset = C3Ch) [Reset = 00000000h]

ANA\_REG\_WU\_MODE\_REG\_LOWV is shown in [Table 5-144](#).

Return to the [Summary Table](#).

**Table 5-144. ANA\_REG\_WU\_MODE\_REG\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved Reads return 0x0 and writes have no effect. 0x00000000 = Functional Reset
6-2	SOP_MODE_LAT_4_0	R	0h	SOP Mode Latched Output Bit2,Bit1,Bit0: 010 SOP_MODE1 SCAN/ATPG 011 SOP_MODE2 DEV/FLED/ORBIT 000 SOP_MODE3 THB 001 SOP_MODE4 FUNC 101 SOP_MODE5 DEV MANAGEMENT 110 SOP_MODE6 DEVBOOT 111 SOP_MODE7 FUNC_SFLASH_RESET Bit4,Bit3: 00 40 MHz 11 50 MHz
1	TEST_MODE_DET_SYNC	R	0h	Latched Output of Test Mode Detect SOP
0	FUNC_TEST_DET_SYNC	R	0h	Latched Output of Functional Test Mode SOP



### 5.2.1.142 ANA\_REG\_WU\_STATUS\_REG\_LOWV Register (Offset = C40h) [Reset = 00000000h]

ANA\_REG\_WU\_STATUS\_REG\_LOWV is shown in [Table 5-145](#).

Return to the [Summary Table](#).

**Table 5-145. ANA\_REG\_WU\_STATUS\_REG\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	Reserved Reads return 0x0 and writes have no effect.
18	VDDS_3P3V_UVDET_LAT	R	0h	Latched Value of 3.3V IO UV Detect 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
17	VDDA_OSC_UVDET_LAT	R	0h	Latched value of UV detect for 1.8V CLK 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
16	APLLVCO18_UVDET_LAT	R	0h	Latched Value of UV Detect for 1.8V VCO 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
15	HVMODE	R	0h	HVMODE Status from VMON 1 = 3.3V VIO 0 = 1.8V VIO
14	LIMP_MODE_STATUS	R	0h	Ref CLK status at Wake-up 0 = REF CLK is present 1 = REF CLK is absent and CPU CLK Switched to RCOSC
13	XTAL_DET_STATUS	R	0h	XTAL Detect status at Wake-up 0 = XTAL absent 1 = XTAL Present
12	RCOSC_CLK_STATUS	R	0h	RCOSC status at Wake-up 0 = RCOSC CLK absent 1 = RCOSC CLK Present
11	REF_CLK_STATUS	R	0h	Ref CLK status at Wake-up 0 = REF CLK absent 1 = REF CLK Present
10	SUPP_OK_VDDD18	R	0h	Can be made spare (TiedLO internally)
9	SUPP_OK_SRAM12	R	0h	Can be made spare (TiedLO internally)
8	VDDA10RF2_UVDET_LAT	R	0h	Latched Value of UV Detect for 1V RF2 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
7	VDDA10RF1_UVDET_LAT	R	0h	Latched Value of UV Detect for 1V RF1 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
6	SUPP_OK_RF10	R	0h	Can be made spare (TiedLO internally)
5	SUPP_OK_IO33	R	0h	Supp Detect output of IO 3.3V 0 = Supply Not detected 1 = Supply Detected
4	SUPP_OK_IO18	R	0h	Supp Detect output of IO 1.8V 0 = Supply Not detected 1 = Supply Detected
3	SUPP_OK_CLK18	R	0h	Supp Detect output of CLK 1.8V 0 = Supply Not detected 1 = Supply Detected
2	VDDA18BB_UV_DET_LAT	R	0h	Latched Value of UV Detect for 1.8V Baseband 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
1	CORE_UVDET_LAT	R	0h	Latched Value of UV Detect 0 = UV Detect Not Triggered 1 = UV Detect has Triggered
0	CORE_OVDET_LAT	R	0h	Latched Value of OV Detect 0 = OV Detect Not Triggered 1 = OV Detect has Triggered

### 5.2.1.143 ANA\_REG\_WU\_SPARE\_OUT\_LOWV Register (Offset = C44h) [Reset = 00000000h]

ANA\_REG\_WU\_SPARE\_OUT\_LOWV is shown in [Table 5-146](#).

Return to the [Summary Table](#).

**Table 5-146. ANA\_REG\_WU\_SPARE\_OUT\_LOWV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved Reads return 0x0 and writes have no effect.
7	CORE_UVDET_LOWV	R	0h	UV Detect of Core Supply-Unlatched
6	CORE_OVDET_LOWV	R	0h	OV Detect of Core Supply-Unlatched
5	INT_OSC_CTRL	R	0h	Internal Oscillator Control
4	SUPPDET_OV_CTRL	R	0h	Supply Detect Override Bit
3	HVMODE	R	0h	Status of VIO supply. 3.3V or 1.8V
2	VDDS18DET	R	0h	Status of 1.8V IO Bias Supply
1	VDDARF_DET	R	0h	Status of 1.3V RF Supply
0	VDDCLK18DET	R	0h	Status of 1.8V CLK Supply

### 5.2.1.144 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-147](#).

Return to the [Summary Table](#).

**Table 5-147. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.1.145 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-148](#).

Return to the [Summary Table](#).

**Table 5-148. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.1.146 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-149](#).

Return to the [Summary Table](#).

**Table 5-149. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D

### 5.2.1.147 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-150](#).

Return to the [Summary Table](#).

**Table 5-150. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.1.148 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-151](#).

Return to the [Summary Table](#).

**Table 5-151. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.1.149 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-152](#).

Return to the [Summary Table](#).

**Table 5-152. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D



### 5.2.1.150 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-153](#).

Return to the [Summary Table](#).

**Table 5-153. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

### 5.2.1.151 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-154](#).

Return to the [Summary Table](#).

**Table 5-154. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.1.152 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-155](#).

Return to the [Summary Table](#).

**Table 5-155. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

### 5.2.1.153 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-156](#).

Return to the [Summary Table](#).

**Table 5-156. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.1.154 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-157](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-157. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

### 5.2.1.155 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-158](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-158. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.1.156 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-159](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-159. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.1.157 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-160](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-160. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.



### 5.2.1.158 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-161](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-161. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.1.159 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-162](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-162. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.1.160 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-163](#).

Return to the [Summary Table](#).

EOI register

**Table 5-163. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.1.161 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-164](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-164. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.1.162 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-165](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-165. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault

### 5.2.1.163 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-166](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-166. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.1.164 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-167](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-167. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

## 5.2.2 MSS\_RCM Registers

Table 5-168 lists the memory-mapped registers for the MSS\_RCM registers. All register offset addresses not listed in Table 5-168 should be considered as reserved locations and the register contents should not be modified.

**Table 5-168. MSS\_RCM Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.2.1</a>
4h	MSS_RST_CAUSE_CLR		<a href="#">Section 5.2.2.2</a>
8h	MSS_RST_STATUS		<a href="#">Section 5.2.2.3</a>
Ch	SYSRST_BY_DBG_RST		<a href="#">Section 5.2.2.4</a>
10h	RST_ASSERTDLY		<a href="#">Section 5.2.2.5</a>
14h	RST2ASSERTDLY		<a href="#">Section 5.2.2.6</a>
18h	RST_WFICHECK		<a href="#">Section 5.2.2.7</a>
24h	MSS_QSPI_CLK_SRC_SEL		<a href="#">Section 5.2.2.8</a>
28h	MSS_RTIA_CLK_SRC_SEL		<a href="#">Section 5.2.2.9</a>
2Ch	MSS_RTIB_CLK_SRC_SEL		<a href="#">Section 5.2.2.10</a>
30h	MSS_RTIC_CLK_SRC_SEL		<a href="#">Section 5.2.2.11</a>
34h	MSS_WDT_CLK_SRC_SEL		<a href="#">Section 5.2.2.12</a>
3Ch	MSS_SPIB_CLK_SRC_SEL		<a href="#">Section 5.2.2.13</a>
40h	MSS_I2C_CLK_SRC_SEL		<a href="#">Section 5.2.2.14</a>
44h	MSS_SCIA_CLK_SRC_SEL		<a href="#">Section 5.2.2.15</a>
48h	MSS_SCIB_CLK_SRC_SEL		<a href="#">Section 5.2.2.16</a>
4Ch	MSS_CPTS_CLK_SRC_SEL		<a href="#">Section 5.2.2.17</a>
50h	MSS_CPSW_CLK_SRC_SEL		<a href="#">Section 5.2.2.18</a>
5Ch	MSS_QSPI_CLK_DIV_VAL		<a href="#">Section 5.2.2.19</a>
60h	MSS_RTIA_CLK_DIV_VAL		<a href="#">Section 5.2.2.20</a>
64h	MSS_RTIB_CLK_DIV_VAL		<a href="#">Section 5.2.2.21</a>
68h	MSS_RTIC_CLK_DIV_VAL		<a href="#">Section 5.2.2.22</a>
6Ch	MSS_WDT_CLK_DIV_VAL		<a href="#">Section 5.2.2.23</a>
74h	MSS_SPIB_CLK_DIV_VAL		<a href="#">Section 5.2.2.24</a>
78h	MSS_I2C_CLK_DIV_VAL		<a href="#">Section 5.2.2.25</a>
7Ch	MSS_SCIA_CLK_DIV_VAL		<a href="#">Section 5.2.2.26</a>
80h	MSS_SCIB_CLK_DIV_VAL		<a href="#">Section 5.2.2.27</a>
84h	MSS_CPTS_CLK_DIV_VAL		<a href="#">Section 5.2.2.28</a>
88h	MSS_CPSW_CLK_DIV_VAL		<a href="#">Section 5.2.2.29</a>
8Ch	MSS_RGMII_CLK_DIV_VAL		<a href="#">Section 5.2.2.30</a>
90h	MSS_MII100_CLK_DIV_VAL		<a href="#">Section 5.2.2.31</a>
94h	MSS_MII10_CLK_DIV_VAL		<a href="#">Section 5.2.2.32</a>
98h	MSS_GPADC_CLK_DIV_VAL		<a href="#">Section 5.2.2.33</a>
A4h	MSS_QSPI_CLK_GATE		<a href="#">Section 5.2.2.34</a>
A8h	MSS_RTIA_CLK_GATE		<a href="#">Section 5.2.2.35</a>
ACh	MSS_RTIB_CLK_GATE		<a href="#">Section 5.2.2.36</a>
B0h	MSS_RTIC_CLK_GATE		<a href="#">Section 5.2.2.37</a>
B4h	MSS_WDT_CLK_GATE		<a href="#">Section 5.2.2.38</a>
BCh	MSS_SPIB_CLK_GATE		<a href="#">Section 5.2.2.39</a>
C0h	MSS_I2C_CLK_GATE		<a href="#">Section 5.2.2.40</a>



**Table 5-168. MSS\_RCM Registers (continued)**

Offset	Acronym	Register Name	Section
C4h	MSS_SCIA_CLK_GATE		<a href="#">Section 5.2.2.41</a>
C8h	MSS_SCIB_CLK_GATE		<a href="#">Section 5.2.2.42</a>
CCh	MSS_CPTS_CLK_GATE		<a href="#">Section 5.2.2.43</a>
D0h	MSS_CPSW_CLK_GATE		<a href="#">Section 5.2.2.44</a>
D4h	MSS_RGMII_CLK_GATE		<a href="#">Section 5.2.2.45</a>
D8h	MSS_MII100_CLK_GATE		<a href="#">Section 5.2.2.46</a>
DCh	MSS_MII10_CLK_GATE		<a href="#">Section 5.2.2.47</a>
E0h	MSS_GPADC_CLK_GATE		<a href="#">Section 5.2.2.48</a>
ECh	MSS_QSPI_CLK_STATUS		<a href="#">Section 5.2.2.49</a>
F0h	MSS_RTIA_CLK_STATUS		<a href="#">Section 5.2.2.50</a>
F4h	MSS_RTIB_CLK_STATUS		<a href="#">Section 5.2.2.51</a>
F8h	MSS_RTIC_CLK_STATUS		<a href="#">Section 5.2.2.52</a>
FCh	MSS_WDT_CLK_STATUS		<a href="#">Section 5.2.2.53</a>
104h	MSS_SPIB_CLK_STATUS		<a href="#">Section 5.2.2.54</a>
108h	MSS_I2C_CLK_STATUS		<a href="#">Section 5.2.2.55</a>
10Ch	MSS_SCIA_CLK_STATUS		<a href="#">Section 5.2.2.56</a>
110h	MSS_SCIB_CLK_STATUS		<a href="#">Section 5.2.2.57</a>
114h	MSS_CPTS_CLK_STATUS		<a href="#">Section 5.2.2.58</a>
118h	MSS_CPSW_CLK_STATUS		<a href="#">Section 5.2.2.59</a>
11Ch	MSS_RGMII_CLK_STATUS		<a href="#">Section 5.2.2.60</a>
120h	MSS_MII100_CLK_STATUS		<a href="#">Section 5.2.2.61</a>
124h	MSS_MII10_CLK_STATUS		<a href="#">Section 5.2.2.62</a>
128h	MSS_GPADC_CLK_STATUS		<a href="#">Section 5.2.2.63</a>
12Ch	MSS_CR5SS_POR_RST_CTRL		<a href="#">Section 5.2.2.64</a>
130h	MSS_CR5SSA_RST_CTRL		<a href="#">Section 5.2.2.65</a>
134h	MSS_CR5SSB_RST_CTRL		<a href="#">Section 5.2.2.66</a>
138h	MSS_CR5A_RST_CTRL		<a href="#">Section 5.2.2.67</a>
13Ch	MSS_CR5B_RST_CTRL		<a href="#">Section 5.2.2.68</a>
140h	MSS_VIMA_RST_CTRL		<a href="#">Section 5.2.2.69</a>
144h	MSS_VIMB_RST_CTRL		<a href="#">Section 5.2.2.70</a>
148h	MSS_CRC_RST_CTRL		<a href="#">Section 5.2.2.71</a>
14Ch	MSS_RTIA_RST_CTRL		<a href="#">Section 5.2.2.72</a>
150h	MSS_RTIB_RST_CTRL		<a href="#">Section 5.2.2.73</a>
154h	MSS_RTIC_RST_CTRL		<a href="#">Section 5.2.2.74</a>
158h	MSS_WDT_RST_CTRL		<a href="#">Section 5.2.2.75</a>
15Ch	MSS_ESM_RST_CTRL		<a href="#">Section 5.2.2.76</a>
160h	MSS_DCCA_RST_CTRL		<a href="#">Section 5.2.2.77</a>
164h	MSS_DCCB_RST_CTRL		<a href="#">Section 5.2.2.78</a>
168h	MSS_DCCC_RST_CTRL		<a href="#">Section 5.2.2.79</a>
16Ch	MSS_DCCD_RST_CTRL		<a href="#">Section 5.2.2.80</a>
170h	MSS_GPIO_RST_CTRL		<a href="#">Section 5.2.2.81</a>
178h	MSS_SPIB_RST_CTRL		<a href="#">Section 5.2.2.82</a>
17Ch	MSS_QSPI_RST_CTRL		<a href="#">Section 5.2.2.83</a>
180h	MSS_PWM1_RST_CTRL		<a href="#">Section 5.2.2.84</a>
184h	MSS_PWM2_RST_CTRL		<a href="#">Section 5.2.2.85</a>

**Table 5-168. MSS\_RCM Registers (continued)**

Offset	Acronym	Register Name	Section
188h	MSS_PWM3_RST_CTRL		<a href="#">Section 5.2.2.86</a>
194h	MSS_I2C_RST_CTRL		<a href="#">Section 5.2.2.87</a>
198h	MSS_SCIA_RST_CTRL		<a href="#">Section 5.2.2.88</a>
19Ch	MSS_SCIB_RST_CTRL		<a href="#">Section 5.2.2.89</a>
1A0h	MSS_EDMA_RST_CTRL		<a href="#">Section 5.2.2.90</a>
1A4h	MSS_INFRA_RST_CTRL		<a href="#">Section 5.2.2.91</a>
1A8h	MSS_CPSW_RST_CTRL		<a href="#">Section 5.2.2.92</a>
1ACh	MSS_GPADC_RST_CTRL		<a href="#">Section 5.2.2.93</a>
1B0h	MSS_DMM_RST_CTRL		<a href="#">Section 5.2.2.94</a>
1B4h	R5_COREA_GATE		<a href="#">Section 5.2.2.95</a>
1B8h	R5_COREB_GATE		<a href="#">Section 5.2.2.96</a>
1BCh	MSS_L2_BANKA_PD_CTRL		<a href="#">Section 5.2.2.97</a>
1C0h	MSS_L2_BANKB_PD_CTRL		<a href="#">Section 5.2.2.98</a>
1C4h	MSS_L2_BANKA_PD_STATUS		<a href="#">Section 5.2.2.99</a>
1C8h	MSS_L2_BANKB_PD_STATUS		<a href="#">Section 5.2.2.100</a>
1CCh	HW_REG0		<a href="#">Section 5.2.2.101</a>
1D0h	HW_REG1		<a href="#">Section 5.2.2.102</a>
1D4h	PREVIOUS_NAME		<a href="#">Section 5.2.2.103</a>
1D8h	HW_REG3		<a href="#">Section 5.2.2.104</a>
1DCh	MSS_CR5F_CLK_SRC_SEL_CTRL		<a href="#">Section 5.2.2.105</a>
1E0h	MSS_CPSW_MII_CLK_SRC_SEL		<a href="#">Section 5.2.2.106</a>
1E4h	MSS_CPSW_MII_CLK_STATUS		<a href="#">Section 5.2.2.107</a>
1E8h	MSS_L2_BANKC_PD_CTRL		<a href="#">Section 5.2.2.108</a>
1ECh	MSS_L2_BANKC_PD_STATUS		<a href="#">Section 5.2.2.109</a>
1F0h	MSS_IP_CLK_CFG		<a href="#">Section 5.2.2.110</a>
400h	HSM_RTIA_CLK_SRC_SEL		<a href="#">Section 5.2.2.111</a>
404h	HSM_WDT_CLK_SRC_SEL		<a href="#">Section 5.2.2.112</a>
408h	HSM_RTC_CLK_SRC_SEL		<a href="#">Section 5.2.2.113</a>
40Ch	HSM_DMTA_CLK_SRC_SEL		<a href="#">Section 5.2.2.114</a>
410h	HSM_DMTB_CLK_SRC_SEL		<a href="#">Section 5.2.2.115</a>
414h	HSM_RTI_CLK_DIV_VAL		<a href="#">Section 5.2.2.116</a>
418h	HSM_WDT_CLK_DIV_VAL		<a href="#">Section 5.2.2.117</a>
41Ch	HSM_RTC_CLK_DIV_VAL		<a href="#">Section 5.2.2.118</a>
420h	HSM_DMTA_CLK_DIV_VAL		<a href="#">Section 5.2.2.119</a>
424h	HSM_DMTB_CLK_DIV_VAL		<a href="#">Section 5.2.2.120</a>
428h	HSM_RTI_CLK_GATE		<a href="#">Section 5.2.2.121</a>
42Ch	HSM_WDT_CLK_GATE		<a href="#">Section 5.2.2.122</a>
430h	HSM_RTC_CLK_GATE		<a href="#">Section 5.2.2.123</a>
434h	HSM_DMTA_CLK_GATE		<a href="#">Section 5.2.2.124</a>
438h	HSM_DMTB_CLK_GATE		<a href="#">Section 5.2.2.125</a>
43Ch	HSM_RTI_CLK_STATUS		<a href="#">Section 5.2.2.126</a>
440h	HSM_WDT_CLK_STATUS		<a href="#">Section 5.2.2.127</a>
444h	HSM_RTC_CLK_STATUS		<a href="#">Section 5.2.2.128</a>
448h	HSM_DMTA_CLK_STATUS		<a href="#">Section 5.2.2.129</a>
44Ch	HSM_DMTB_CLK_STATUS		<a href="#">Section 5.2.2.130</a>

**Table 5-168. MSS\_RCM Registers (continued)**

Offset	Acronym	Register Name	Section
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.2.131</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.2.132</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.2.133</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.2.134</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.2.135</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.2.136</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.2.137</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.2.138</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.2.139</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.2.140</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.2.141</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.2.142</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.2.143</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.2.144</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.2.145</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.2.146</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.2.147</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.2.148</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.2.149</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.2.150</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.2.151</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-169](#) shows the codes that are used for access types in this section.

**Table 5-169. MSS\_RCM Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.2.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-170](#).

Return to the [Summary Table](#).

PID register

**Table 5-170. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.2.2 MSS\_RST\_CAUSE\_CLR Register (Offset = 4h) [Reset = 00000000h]

MSS\_RST\_CAUSE\_CLR is shown in [Table 5-171](#).

Return to the [Summary Table](#).

**Table 5-171. MSS\_RST\_CAUSE\_CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	clr	R/W	0h	Write pulse bit field: Clear bit for rst cause register (writing '111' will clear the rst cause register)

### 5.2.2.3 MSS\_RST\_STATUS Register (Offset = 8h) [Reset = 00000003h]

MSS\_RST\_STATUS is shown in [Table 5-172](#).

Return to the [Summary Table](#).

**Table 5-172. MSS\_RST\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	cause	R	3h	Has the status because of which reset has happened. Bit0: POR Reset Bit1: Warm Reset Bit2: STC Reset Bit3 Reset for CR5A and MSS_CR5A_VIM using MSS_RCM::MSS_CR5SSA_RST_CTRL Bit4: RESERVED Bit5: Reset for CR5A only using MSS_RCM::MSS_CR5A_RST_CTRL Bit6: RESERVED Bit7: Reset for CR5A and MSS_CR5A_VIM caused because of reset request by debugger in CR5A Bit8: RESERVED Bit9: Reset for CR5SS by the RESET FSM using MSS_CTRL::R5_CONTROL_RESET_FSM_TRIGGER

### 5.2.2.4 SYSRST\_BY\_DBG\_RST Register (Offset = Ch) [Reset = 0000XXXXh]

SYSRST\_BY\_DBG\_RST is shown in [Table 5-173](#).

Return to the [Summary Table](#).

**Table 5-173. SYSRST\_BY\_DBG\_RST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	r5b	R/W	0h	RESERVED: Dont Use
15-3	RESERVED	R/W	0h	
2-0	r5a	R/W	0h	writing '111' will block debug reset request from CR5A toggling globally reset for CR5A

### 5.2.2.5 RST\_ASSERDLY Register (Offset = 10h) [Reset = 000000Fh]

RST\_ASSERDLY is shown in [Table 5-174](#).

Return to the [Summary Table](#).

**Table 5-174. RST\_ASSERDLY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	common	R/W	Fh	Value decides number of cycles reset should be asserted for CR5SS related resets



### 5.2.2.6 RST2ASSERTDLY Register (Offset = 14h) [Reset = 00000000h]

RST2ASSERTDLY is shown in [Table 5-175](#).

Return to the [Summary Table](#).

**Table 5-175. RST2ASSERTDLY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	r5b	R/W	0h	RESERVED: Dont Use
23-16	r5a	R/W	0h	Value decides number of cycles should be held before asserting reset for r5ss local reset for CR5A
15-8	r5ssb	R/W	0h	RESERVED: Dont Use
7-0	r5ssa	R/W	0h	Value decides number of cycles should be held before asserting reset for r5ss global reset for CR5A.

### 5.2.2.7 RST\_WFICHECK Register (Offset = 18h) [Reset = 07XXXXXXh]

RST\_WFICHECK is shown in [Table 5-176](#).

Return to the [Summary Table](#).

**Table 5-176. RST\_WFICHECK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	r5b	R/W	7h	writing '000' will disable check for WFI before local reset assertion of CR5A
23-19	RESERVED	R/W	0h	
18-16	r5a	R/W	7h	writing '000' will disable check for WFI before local reset assertion of CR5A
15-11	RESERVED	R/W	0h	
10-8	r5ssb	R/W	7h	RESERVED: Dont Use
7-3	RESERVED	R/W	0h	
2-0	r5ssa	R/W	7h	writing '000' will disable check for WFI before global reset assertion of CR5A

### 5.2.2.8 MSS\_QSPI\_CLK\_SRC\_SEL Register (Offset = 24h) [Reset = 00000000h]

MSS\_QSPI\_CLK\_SRC\_SEL is shown in [Table 5-177](#).

Return to the [Summary Table](#).

**Table 5-177. MSS\_QSPI\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for QSPI. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.9 MSS\_RTIA\_CLK\_SRC\_SEL Register (Offset = 28h) [Reset = 00000000h]

MSS\_RTIA\_CLK\_SRC\_SEL is shown in [Table 5-178](#).

Return to the [Summary Table](#).

**Table 5-178. MSS\_RTIA\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for RTIA. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.10 MSS\_RTIB\_CLK\_SRC\_SEL Register (Offset = 2Ch) [Reset = 00000000h]

MSS\_RTIB\_CLK\_SRC\_SEL is shown in [Table 5-179](#).

Return to the [Summary Table](#).

**Table 5-179. MSS\_RTIB\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for RTIB. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.11 MSS\_RTIC\_CLK\_SRC\_SEL Register (Offset = 30h) [Reset = 00000000h]

MSS\_RTIC\_CLK\_SRC\_SEL is shown in [Table 5-180](#).

Return to the [Summary Table](#).

**Table 5-180. MSS\_RTIC\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for RTIC. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.12 MSS\_WDT\_CLK\_SRC\_SEL Register (Offset = 34h) [Reset = 00000000h]

MSS\_WDT\_CLK\_SRC\_SEL is shown in [Table 5-181](#).

Return to the [Summary Table](#).

**Table 5-181. MSS\_WDT\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for WDT. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.13 MSS\_SPIB\_CLK\_SRC\_SEL Register (Offset = 3Ch) [Reset = 00000000h]

MSS\_SPIB\_CLK\_SRC\_SEL is shown in [Table 5-182](#).

Return to the [Summary Table](#).

**Table 5-182. MSS\_SPIB\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for SPIB. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference



#### 5.2.2.14 MSS\_I2C\_CLK\_SRC\_SEL Register (Offset = 40h) [Reset = 00000000h]

MSS\_I2C\_CLK\_SRC\_SEL is shown in [Table 5-183](#).

Return to the [Summary Table](#).

**Table 5-183. MSS\_I2C\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for I2C. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.15 MSS\_SCIA\_CLK\_SRC\_SEL Register (Offset = 44h) [Reset = 00000000h]

MSS\_SCIA\_CLK\_SRC\_SEL is shown in [Table 5-184](#).

Return to the [Summary Table](#).

**Table 5-184. MSS\_SCIA\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for SCIA. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.16 MSS\_SCIB\_CLK\_SRC\_SEL Register (Offset = 48h) [Reset = 00000000h]

MSS\_SCIB\_CLK\_SRC\_SEL is shown in [Table 5-185](#).

Return to the [Summary Table](#).

**Table 5-185. MSS\_SCIB\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for SCIB. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.17 MSS\_CPTS\_CLK\_SRC\_SEL Register (Offset = 4Ch) [Reset = 00000000h]

MSS\_CPTS\_CLK\_SRC\_SEL is shown in [Table 5-186](#).

Return to the [Summary Table](#).

**Table 5-186. MSS\_CPTS\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for CPTS. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.18 MSS\_CPSW\_CLK\_SRC\_SEL Register (Offset = 50h) [Reset = 00000000h]

MSS\_CPSW\_CLK\_SRC\_SEL is shown in [Table 5-187](#).

Return to the [Summary Table](#).

**Table 5-187. MSS\_CPSW\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for CPSW. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.19 MSS\_QSPI\_CLK\_DIV\_VAL Register (Offset = 5Ch) [Reset = 00000000h]

MSS\_QSPI\_CLK\_DIV\_VAL is shown in [Table 5-188](#).

Return to the [Summary Table](#).

**Table 5-188. MSS\_QSPI\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value QSPI selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.20 MSS\_RTIA\_CLK\_DIV\_VAL Register (Offset = 60h) [Reset = 00000000h]

MSS\_RTIA\_CLK\_DIV\_VAL is shown in [Table 5-189](#).

Return to the [Summary Table](#).

**Table 5-189. MSS\_RTIA\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value RTIA selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.21 MSS\_RTIB\_CLK\_DIV\_VAL Register (Offset = 64h) [Reset = 00000000h]

MSS\_RTIB\_CLK\_DIV\_VAL is shown in [Table 5-190](#).

Return to the [Summary Table](#).

**Table 5-190. MSS\_RTIB\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value RTIB selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference



### 5.2.2.22 MSS\_RTIC\_CLK\_DIV\_VAL Register (Offset = 68h) [Reset = 00000000h]

MSS\_RTIC\_CLK\_DIV\_VAL is shown in [Table 5-191](#).

Return to the [Summary Table](#).

**Table 5-191. MSS\_RTIC\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value RTIC selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.23 MSS\_WDT\_CLK\_DIV\_VAL Register (Offset = 6Ch) [Reset = 00000000h]

MSS\_WDT\_CLK\_DIV\_VAL is shown in [Table 5-192](#).

Return to the [Summary Table](#).

**Table 5-192. MSS\_WDT\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value WDT selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

#### 5.2.2.24 MSS\_SPIB\_CLK\_DIV\_VAL Register (Offset = 74h) [Reset = 00000000h]

MSS\_SPIB\_CLK\_DIV\_VAL is shown in [Table 5-193](#).

Return to the [Summary Table](#).

**Table 5-193. MSS\_SPIB\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value SPIB selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.25 MSS\_I2C\_CLK\_DIV\_VAL Register (Offset = 78h) [Reset = 00000000h]

MSS\_I2C\_CLK\_DIV\_VAL is shown in [Table 5-194](#).

Return to the [Summary Table](#).

**Table 5-194. MSS\_I2C\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value I2C selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.26 MSS\_SCIA\_CLK\_DIV\_VAL Register (Offset = 7Ch) [Reset = 00000000h]

MSS\_SCIA\_CLK\_DIV\_VAL is shown in [Table 5-195](#).

Return to the [Summary Table](#).

**Table 5-195. MSS\_SCIA\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value SCIA selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.27 MSS\_SCIB\_CLK\_DIV\_VAL Register (Offset = 80h) [Reset = 00000000h]

MSS\_SCIB\_CLK\_DIV\_VAL is shown in [Table 5-196](#).

Return to the [Summary Table](#).

**Table 5-196. MSS\_SCIB\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value SCIB selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.28 MSS\_CPTS\_CLK\_DIV\_VAL Register (Offset = 84h) [Reset = 00000000h]

MSS\_CPTS\_CLK\_DIV\_VAL is shown in [Table 5-197](#).

Return to the [Summary Table](#).

**Table 5-197. MSS\_CPTS\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value CPTS selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.29 MSS\_CPSW\_CLK\_DIV\_VAL Register (Offset = 88h) [Reset = 00000000h]

MSS\_CPSW\_CLK\_DIV\_VAL is shown in [Table 5-198](#).

Return to the [Summary Table](#).

**Table 5-198. MSS\_CPSW\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value CPSW selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference



### 5.2.2.30 MSS\_RGMII\_CLK\_DIV\_VAL Register (Offset = 8Ch) [Reset = 00000000h]

MSS\_RGMII\_CLK\_DIV\_VAL is shown in [Table 5-199](#).

Return to the [Summary Table](#).

**Table 5-199. MSS\_RGMII\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value RGMII selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.31 MSS\_MII100\_CLK\_DIV\_VAL Register (Offset = 90h) [Reset = 00000000h]

MSS\_MII100\_CLK\_DIV\_VAL is shown in [Table 5-200](#).

Return to the [Summary Table](#).

**Table 5-200. MSS\_MII100\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value MII100 selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.32 MSS\_MII10\_CLK\_DIV\_VAL Register (Offset = 94h) [Reset = 00000000h]

MSS\_MII10\_CLK\_DIV\_VAL is shown in [Table 5-201](#).

Return to the [Summary Table](#).

**Table 5-201. MSS\_MII10\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	clkdivr	R/W	0h	Divider value MII10 selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.33 MSS\_GPADC\_CLK\_DIV\_VAL Register (Offset = 98h) [Reset = 00000000h]

MSS\_GPADC\_CLK\_DIV\_VAL is shown in [Table 5-202](#).

Return to the [Summary Table](#).

**Table 5-202. MSS\_GPADC\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-0	clkdivr	R/W	0h	Divider value GPADC selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.34 MSS\_QSPI\_CLK\_GATE Register (Offset = A4h) [Reset = 00000000h]

MSS\_QSPI\_CLK\_GATE is shown in [Table 5-203](#).

Return to the [Summary Table](#).

**Table 5-203. MSS\_QSPI\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for QSPI

### 5.2.2.35 MSS\_RTIA\_CLK\_GATE Register (Offset = A8h) [Reset = 00000000h]

MSS\_RTIA\_CLK\_GATE is shown in [Table 5-204](#).

Return to the [Summary Table](#).

**Table 5-204. MSS\_RTIA\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for RTIA

### 5.2.2.36 MSS\_RTIB\_CLK\_GATE Register (Offset = ACh) [Reset = 00000000h]

MSS\_RTIB\_CLK\_GATE is shown in [Table 5-205](#).

Return to the [Summary Table](#).

**Table 5-205. MSS\_RTIB\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for RTIB

### 5.2.2.37 MSS\_RTIC\_CLK\_GATE Register (Offset = B0h) [Reset = 00000000h]

MSS\_RTIC\_CLK\_GATE is shown in [Table 5-206](#).

Return to the [Summary Table](#).

**Table 5-206. MSS\_RTIC\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for RTIC



### 5.2.2.38 MSS\_WDT\_CLK\_GATE Register (Offset = B4h) [Reset = 00000000h]

MSS\_WDT\_CLK\_GATE is shown in [Table 5-207](#).

Return to the [Summary Table](#).

**Table 5-207. MSS\_WDT\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for WDT

### 5.2.2.39 MSS\_SPIB\_CLK\_GATE Register (Offset = BCh) [Reset = 00000000h]

MSS\_SPIB\_CLK\_GATE is shown in [Table 5-208](#).

Return to the [Summary Table](#).

**Table 5-208. MSS\_SPIB\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for SPIB

### 5.2.2.40 MSS\_I2C\_CLK\_GATE Register (Offset = C0h) [Reset = 00000000h]

MSS\_I2C\_CLK\_GATE is shown in [Table 5-209](#).

Return to the [Summary Table](#).

**Table 5-209. MSS\_I2C\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for I2C

#### 5.2.2.41 MSS\_SCIA\_CLK\_GATE Register (Offset = C4h) [Reset = 00000000h]

MSS\_SCIA\_CLK\_GATE is shown in [Table 5-210](#).

Return to the [Summary Table](#).

**Table 5-210. MSS\_SCIA\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for SCIA

#### 5.2.2.42 MSS\_SCIB\_CLK\_GATE Register (Offset = C8h) [Reset = 00000000h]

MSS\_SCIB\_CLK\_GATE is shown in [Table 5-211](#).

Return to the [Summary Table](#).

**Table 5-211. MSS\_SCIB\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for SCIB

### 5.2.2.43 MSS\_CPTS\_CLK\_GATE Register (Offset = CCh) [Reset = 00000000h]

MSS\_CPTS\_CLK\_GATE is shown in [Table 5-212](#).

Return to the [Summary Table](#).

**Table 5-212. MSS\_CPTS\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for CPTS

#### 5.2.2.44 MSS\_CPSW\_CLK\_GATE Register (Offset = D0h) [Reset = 00000000h]

MSS\_CPSW\_CLK\_GATE is shown in [Table 5-213](#).

Return to the [Summary Table](#).

**Table 5-213. MSS\_CPSW\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for CPSW

### 5.2.2.45 MSS\_RGMII\_CLK\_GATE Register (Offset = D4h) [Reset = 00000000h]

MSS\_RGMII\_CLK\_GATE is shown in [Table 5-214](#).

Return to the [Summary Table](#).

**Table 5-214. MSS\_RGMII\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for RGMII



### 5.2.2.46 MSS\_MII100\_CLK\_GATE Register (Offset = D8h) [Reset = 00000000h]

MSS\_MII100\_CLK\_GATE is shown in [Table 5-215](#).

Return to the [Summary Table](#).

**Table 5-215. MSS\_MII100\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for MII100

### 5.2.2.47 MSS\_MII10\_CLK\_GATE Register (Offset = DCh) [Reset = 00000000h]

MSS\_MII10\_CLK\_GATE is shown in [Table 5-216](#).

Return to the [Summary Table](#).

**Table 5-216. MSS\_MII10\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for MII10

### 5.2.2.48 MSS\_GPADC\_CLK\_GATE Register (Offset = E0h) [Reset = 00000000h]

MSS\_GPADC\_CLK\_GATE is shown in [Table 5-217](#).

Return to the [Summary Table](#).

**Table 5-217. MSS\_GPADC\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for MSS GPADC

### 5.2.2.49 MSS\_QSPI\_CLK\_STATUS Register (Offset = ECh) [Reset = 00000001h]

MSS\_QSPI\_CLK\_STATUS is shown in [Table 5-218](#).

Return to the [Summary Table](#).

**Table 5-218. MSS\_QSPI\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for QSPI
7-0	clkinuse	R	1h	Status shows the source clock slected for QSPI

### 5.2.2.50 MSS\_RTIA\_CLK\_STATUS Register (Offset = F0h) [Reset = 00000001h]

MSS\_RTIA\_CLK\_STATUS is shown in [Table 5-219](#).

Return to the [Summary Table](#).

**Table 5-219. MSS\_RTIA\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for RTIA
7-0	clkinuse	R	1h	Status shows the source clock slected for RTIA

### 5.2.2.51 MSS\_RTIB\_CLK\_STATUS Register (Offset = F4h) [Reset = 00000001h]

MSS\_RTIB\_CLK\_STATUS is shown in [Table 5-220](#).

Return to the [Summary Table](#).

**Table 5-220. MSS\_RTIB\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for RTIB
7-0	clkinuse	R	1h	Status shows the source clock slected for RTIB

### 5.2.2.52 MSS\_RTIC\_CLK\_STATUS Register (Offset = F8h) [Reset = 00000001h]

MSS\_RTIC\_CLK\_STATUS is shown in [Table 5-221](#).

Return to the [Summary Table](#).

**Table 5-221. MSS\_RTIC\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for RTIC
7-0	clkinuse	R	1h	Status shows the source clock slected for RTIC

### 5.2.2.53 MSS\_WDT\_CLK\_STATUS Register (Offset = FCh) [Reset = 00000001h]

MSS\_WDT\_CLK\_STATUS is shown in [Table 5-222](#).

Return to the [Summary Table](#).

**Table 5-222. MSS\_WDT\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for WDT
7-0	clkinuse	R	1h	Status shows the source clock slected for WDT



### 5.2.2.54 MSS\_SPIB\_CLK\_STATUS Register (Offset = 104h) [Reset = 00000001h]

MSS\_SPIB\_CLK\_STATUS is shown in [Table 5-223](#).

Return to the [Summary Table](#).

**Table 5-223. MSS\_SPIB\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for SPIB
7-0	clkinuse	R	1h	Status shows the source clock slected for SPIB

### 5.2.2.55 MSS\_I2C\_CLK\_STATUS Register (Offset = 108h) [Reset = 00000001h]

MSS\_I2C\_CLK\_STATUS is shown in [Table 5-224](#).

Return to the [Summary Table](#).

**Table 5-224. MSS\_I2C\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for I2C
7-0	clkinuse	R	1h	Status shows the source clock slected for I2C

### 5.2.2.56 MSS\_SCIA\_CLK\_STATUS Register (Offset = 10Ch) [Reset = 00000001h]

MSS\_SCIA\_CLK\_STATUS is shown in [Table 5-225](#).

Return to the [Summary Table](#).

**Table 5-225. MSS\_SCIA\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for SCIA
7-0	clkinuse	R	1h	Status shows the source clock slected for SCIA

### 5.2.2.57 MSS\_SCIB\_CLK\_STATUS Register (Offset = 110h) [Reset = 00000001h]

MSS\_SCIB\_CLK\_STATUS is shown in [Table 5-226](#).

Return to the [Summary Table](#).

**Table 5-226. MSS\_SCIB\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for SCIB
7-0	clkinuse	R	1h	Status shows the source clock slected for SCIB

### 5.2.2.58 MSS\_CPTS\_CLK\_STATUS Register (Offset = 114h) [Reset = 00000001h]

MSS\_CPTS\_CLK\_STATUS is shown in [Table 5-227](#).

Return to the [Summary Table](#).

**Table 5-227. MSS\_CPTS\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for CPTS
7-0	clkinuse	R	1h	Status shows the source clock slected for CPTS

### 5.2.2.59 MSS\_CPSW\_CLK\_STATUS Register (Offset = 118h) [Reset = 00000001h]

MSS\_CPSW\_CLK\_STATUS is shown in [Table 5-228](#).

Return to the [Summary Table](#).

**Table 5-228. MSS\_CPSW\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for CPSW
7-0	clkinuse	R	1h	Status shows the source clock slected for CPSW

### 5.2.2.60 MSS\_RGMII\_CLK\_STATUS Register (Offset = 11Ch) [Reset = 000001XXh]

MSS\_RGMII\_CLK\_STATUS is shown in [Table 5-229](#).

Return to the [Summary Table](#).

**Table 5-229. MSS\_RGMII\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	1h	Status shows the current divider value choosen for RGMII
7-0	RESERVED	R	0h	

### 5.2.2.61 MSS\_MII100\_CLK\_STATUS Register (Offset = 120h) [Reset = 000001XXh]

MSS\_MII100\_CLK\_STATUS is shown in [Table 5-230](#).

Return to the [Summary Table](#).

**Table 5-230. MSS\_MII100\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	1h	Status shows the current divider value choosen for MII100
7-0	RESERVED	R	0h	



### 5.2.2.62 MSS\_MII10\_CLK\_STATUS Register (Offset = 124h) [Reset = 000001XXh]

MSS\_MII10\_CLK\_STATUS is shown in [Table 5-231](#).

Return to the [Summary Table](#).

**Table 5-231. MSS\_MII10\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	1h	Status shows the current divider value choosen for MII10
7-0	RESERVED	R	0h	

### 5.2.2.63 MSS\_GPADC\_CLK\_STATUS Register (Offset = 128h) [Reset = 000000XXh]

MSS\_GPADC\_CLK\_STATUS is shown in [Table 5-232](#).

Return to the [Summary Table](#).

**Table 5-232. MSS\_GPADC\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for GPADC
7-0	RESERVED	R	0h	

### 5.2.2.64 MSS\_CR5SS\_POR\_RST\_CTRL Register (Offset = 12Ch) [Reset = 00000000h]

MSS\_CR5SS\_POR\_RST\_CTRL is shown in [Table 5-233](#).

Return to the [Summary Table](#).

**Table 5-233. MSS\_CR5SS\_POR\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. write pulse bit field: writing '111' will assert por reset to R5SS

### 5.2.2.65 MSS\_CR5SSA\_RST\_CTRL Register (Offset = 130h) [Reset = 00000000h]

MSS\_CR5SSA\_RST\_CTRL is shown in [Table 5-234](#).

Return to the [Summary Table](#).

**Table 5-234. MSS\_CR5SSA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. write pulse bit field: writing '111' will reset CR5A and MSS_CR5A_VIM

### 5.2.2.66 MSS\_CR5SSB\_RST\_CTRL Register (Offset = 134h) [Reset = 00000000h]

MSS\_CR5SSB\_RST\_CTRL is shown in [Table 5-235](#).

Return to the [Summary Table](#).

**Table 5-235. MSS\_CR5SSB\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	RESERVED: Dont Use

### 5.2.2.67 MSS\_CR5A\_RST\_CTRL Register (Offset = 138h) [Reset = 00000000h]

MSS\_CR5A\_RST\_CTRL is shown in [Table 5-236](#).

Return to the [Summary Table](#).

**Table 5-236. MSS\_CR5A\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. write pulse bit field: writing '111' will reset CR5A only

### 5.2.2.68 MSS\_CR5B\_RST\_CTRL Register (Offset = 13Ch) [Reset = 00000000h]

MSS\_CR5B\_RST\_CTRL is shown in [Table 5-237](#).

Return to the [Summary Table](#).

**Table 5-237. MSS\_CR5B\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	RESERVED: Dont Use

### 5.2.2.69 MSS\_VIMA\_RST\_CTRL Register (Offset = 140h) [Reset = 00000000h]

MSS\_VIMA\_RST\_CTRL is shown in [Table 5-238](#).

Return to the [Summary Table](#).

**Table 5-238. MSS\_VIMA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS_CR5A_VIM



### 5.2.2.70 MSS\_VIMB\_RST\_CTRL Register (Offset = 144h) [Reset = 00000000h]

MSS\_VIMB\_RST\_CTRL is shown in [Table 5-239](#).

Return to the [Summary Table](#).

**Table 5-239. MSS\_VIMB\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	RESERVED: Dont Use

### 5.2.2.71 MSS\_CRC\_RST\_CTRL Register (Offset = 148h) [Reset = 00000000h]

MSS\_CRC\_RST\_CTRL is shown in [Table 5-240](#).

Return to the [Summary Table](#).

**Table 5-240. MSS\_CRC\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MCRC

### 5.2.2.72 MSS\_RTIA\_RST\_CTRL Register (Offset = 14Ch) [Reset = 00000000h]

MSS\_RTIA\_RST\_CTRL is shown in [Table 5-241](#).

Return to the [Summary Table](#).

**Table 5-241. MSS\_RTIA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset RTIA

### 5.2.2.73 MSS\_RTIB\_RST\_CTRL Register (Offset = 150h) [Reset = 00000000h]

MSS\_RTIB\_RST\_CTRL is shown in [Table 5-242](#).

Return to the [Summary Table](#).

**Table 5-242. MSS\_RTIB\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset RTIB

### 5.2.2.74 MSS\_RTIC\_RST\_CTRL Register (Offset = 154h) [Reset = 00000000h]

MSS\_RTIC\_RST\_CTRL is shown in [Table 5-243](#).

Return to the [Summary Table](#).

**Table 5-243. MSS\_RTIC\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset RTIC

### 5.2.2.75 MSS\_WDT\_RST\_CTRL Register (Offset = 158h) [Reset = 00000000h]

MSS\_WDT\_RST\_CTRL is shown in [Table 5-244](#).

Return to the [Summary Table](#).

**Table 5-244. MSS\_WDT\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset WDT

### 5.2.2.76 MSS\_ESM\_RST\_CTRL Register (Offset = 15Ch) [Reset = 00000000h]

MSS\_ESM\_RST\_CTRL is shown in [Table 5-245](#).

Return to the [Summary Table](#).

**Table 5-245. MSS\_ESM\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset ESM

### 5.2.2.77 MSS\_DCCA\_RST\_CTRL Register (Offset = 160h) [Reset = 00000000h]

MSS\_DCCA\_RST\_CTRL is shown in [Table 5-246](#).

Return to the [Summary Table](#).

**Table 5-246. MSS\_DCCA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset DCCA



### 5.2.2.78 MSS\_DCCB\_RST\_CTRL Register (Offset = 164h) [Reset = 00000000h]

MSS\_DCCB\_RST\_CTRL is shown in [Table 5-247](#).

Return to the [Summary Table](#).

**Table 5-247. MSS\_DCCB\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset DCCB

### 5.2.2.79 MSS\_DCCC\_RST\_CTRL Register (Offset = 168h) [Reset = 00000000h]

MSS\_DCCC\_RST\_CTRL is shown in [Table 5-248](#).

Return to the [Summary Table](#).

**Table 5-248. MSS\_DCCC\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset DCCC

### 5.2.2.80 MSS\_DCCD\_RST\_CTRL Register (Offset = 16Ch) [Reset = 00000000h]

MSS\_DCCD\_RST\_CTRL is shown in [Table 5-249](#).

Return to the [Summary Table](#).

**Table 5-249. MSS\_DCCD\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset DCCD

### 5.2.2.81 MSS\_GPIO\_RST\_CTRL Register (Offset = 170h) [Reset = 00000000h]

MSS\_GPIO\_RST\_CTRL is shown in [Table 5-250](#).

Return to the [Summary Table](#).

**Table 5-250. MSS\_GPIO\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset GIO

### 5.2.2.82 MSS\_SPIB\_RST\_CTRL Register (Offset = 178h) [Reset = 00000000h]

MSS\_SPIB\_RST\_CTRL is shown in [Table 5-251](#).

Return to the [Summary Table](#).

**Table 5-251. MSS\_SPIB\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset SPIB

### 5.2.2.83 MSS\_QSPI\_RST\_CTRL Register (Offset = 17Ch) [Reset = 00000000h]

MSS\_QSPI\_RST\_CTRL is shown in [Table 5-252](#).

Return to the [Summary Table](#).

**Table 5-252. MSS\_QSPI\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset QSPI

### 5.2.2.84 MSS\_PWM1\_RST\_CTRL Register (Offset = 180h) [Reset = 00000000h]

MSS\_PWM1\_RST\_CTRL is shown in [Table 5-253](#).

Return to the [Summary Table](#).

**Table 5-253. MSS\_PWM1\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset EPWM1

### 5.2.2.85 MSS\_PWM2\_RST\_CTRL Register (Offset = 184h) [Reset = 00000000h]

MSS\_PWM2\_RST\_CTRL is shown in [Table 5-254](#).

Return to the [Summary Table](#).

**Table 5-254. MSS\_PWM2\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset EPWM2



### 5.2.2.86 MSS\_PWM3\_RST\_CTRL Register (Offset = 188h) [Reset = 00000000h]

MSS\_PWM3\_RST\_CTRL is shown in [Table 5-255](#).

Return to the [Summary Table](#).

**Table 5-255. MSS\_PWM3\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset EPWM3

### 5.2.2.87 MSS\_I2C\_RST\_CTRL Register (Offset = 194h) [Reset = 00000000h]

MSS\_I2C\_RST\_CTRL is shown in [Table 5-256](#).

Return to the [Summary Table](#).

**Table 5-256. MSS\_I2C\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset I2C

### 5.2.2.88 MSS\_SCIA\_RST\_CTRL Register (Offset = 198h) [Reset = 00000000h]

MSS\_SCIA\_RST\_CTRL is shown in [Table 5-257](#).

Return to the [Summary Table](#).

**Table 5-257. MSS\_SCIA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset SCIA

### 5.2.2.89 MSS\_SCIB\_RST\_CTRL Register (Offset = 19Ch) [Reset = 00000000h]

MSS\_SCIB\_RST\_CTRL is shown in [Table 5-258](#).

Return to the [Summary Table](#).

**Table 5-258. MSS\_SCIB\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset SCIB

### 5.2.2.90 MSS\_EDMA\_RST\_CTRL Register (Offset = 1A0h) [Reset = 00XXXXXXh]

MSS\_EDMA\_RST\_CTRL is shown in [Table 5-259](#).

Return to the [Summary Table](#).

**Table 5-259. MSS\_EDMA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	tp tcb0_assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS_TPTCB0
23-19	RESERVED	R/W	0h	
18-16	tpccb_assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS_TPCCB
15	RESERVED	R/W	0h	
14-12	tp tca1_assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS_TPTCA1
11	RESERVED	R/W	0h	
10-8	tp tca0_assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS_TPTCA0
7	RESERVED	R/W	0h	
6-4	tpcca_assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS_TPCCA
3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset EDMA

### 5.2.2.91 MSS\_INFRA\_RST\_CTRL Register (Offset = 1A4h) [Reset = 00000000h]

MSS\_INFRA\_RST\_CTRL is shown in [Table 5-260](#).

Return to the [Summary Table](#).

**Table 5-260. MSS\_INFRA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS INFRA

### 5.2.2.92 MSS\_CPSW\_RST\_CTRL Register (Offset = 1A8h) [Reset = 00000000h]

MSS\_CPSW\_RST\_CTRL is shown in [Table 5-261](#).

Return to the [Summary Table](#).

**Table 5-261. MSS\_CPSW\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS CPSW

### 5.2.2.93 MSS\_GPADC\_RST\_CTRL Register (Offset = 1ACh) [Reset = 00000000h]

MSS\_GPADC\_RST\_CTRL is shown in [Table 5-262](#).

Return to the [Summary Table](#).

**Table 5-262. MSS\_GPADC\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS GPADC



### 5.2.2.94 MSS\_DMM\_RST\_CTRL Register (Offset = 1B0h) [Reset = 00000000h]

MSS\_DMM\_RST\_CTRL is shown in [Table 5-263](#).

Return to the [Summary Table](#).

**Table 5-263. MSS\_DMM\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of the Device/IP before configuring. Writing '111' will reset MSS DMMA/B

### 5.2.2.95 R5\_COREA\_GATE Register (Offset = 1B4h) [Reset = 00000000h]

R5\_COREA\_GATE is shown in [Table 5-264](#).

Return to the [Summary Table](#).

**Table 5-264. R5\_COREA\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	clkgate	R/W	0h	writing '111' will gate clock to CR5A related peripherals inside Cortexr5ss

### 5.2.2.96 R5\_COREB\_GATE Register (Offset = 1B8h) [Reset = 00000000h]

R5\_COREB\_GATE is shown in [Table 5-265](#).

Return to the [Summary Table](#).

**Table 5-265. R5\_COREB\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	clkgate	R/W	0h	writing '111' will gate clock to CR5B related peripherals inside Cortexr5ss

### 5.2.2.97 MSS\_L2\_BANKA\_PD\_CTRL Register (Offset = 1BCh) [Reset = 000007XXh]

MSS\_L2\_BANKA\_PD\_CTRL is shown in [Table 5-266](#).

Return to the [Summary Table](#).

**Table 5-266. MSS\_L2\_BANKA\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW control for power signal 'AGOODIN' for MSS_L2_BANKA
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW control for power signal 'AONIN' for MSS_L2_BANKA
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW control for power signal 'ISO' for MSS_L2_BANKA

### 5.2.2.98 MSS\_L2\_BANKB\_PD\_CTRL Register (Offset = 1C0h) [Reset = 000007XXh]

MSS\_L2\_BANKB\_PD\_CTRL is shown in [Table 5-267](#).

Return to the [Summary Table](#).

**Table 5-267. MSS\_L2\_BANKB\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW control for power signal 'AGOODIN' for MSS_L2_BANKB
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW control for power signal 'AONIN' for MSS_L2_BANKB
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW control for power signal 'ISO' for MSS_L2_BANKB

### 5.2.2.99 MSS\_L2\_BANKA\_PD\_STATUS Register (Offset = 1C4h) [Reset = 00000003h]

MSS\_L2\_BANKA\_PD\_STATUS is shown in [Table 5-268](#).

Return to the [Summary Table](#).

**Table 5-268. MSS\_L2\_BANKA\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	agoodout	R	1h	SW status indicating the 'pgoodin' of MSS_L2_BANKA
0	aonout	R	1h	SW status indicating the 'ponin' of MSS_L2_BANKA

### 5.2.2.100 MSS\_L2\_BANKB\_PD\_STATUS Register (Offset = 1C8h) [Reset = 00000003h]

MSS\_L2\_BANKB\_PD\_STATUS is shown in [Table 5-269](#).

Return to the [Summary Table](#).

**Table 5-269. MSS\_L2\_BANKB\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	agoodout	R	1h	SW status indicating the 'pgoodin' of MSS_L2_BANKB
0	aonout	R	1h	SW status indicating the 'ponin' of MSS_L2_BANKB

### 5.2.2.101 HW\_REG0 Register (Offset = 1CCh) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-270](#).

Return to the [Summary Table](#).

**Table 5-270. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser



### 5.2.2.102 HW\_REG1 Register (Offset = 1D0h) [Reset = 00000000h]

HW\_REG1 is shown in [Table 5-271](#).

Return to the [Summary Table](#).

**Table 5-271. HW\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

### 5.2.2.103 PREVIOUS\_NAME Register (Offset = 1D4h) [Reset = 00000000h]

PREVIOUS\_NAME is shown in [Table 5-272](#).

Return to the [Summary Table](#).

**Table 5-272. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

### 5.2.2.104 HW\_REG3 Register (Offset = 1D8h) [Reset = 00000000h]

HW\_REG3 is shown in [Table 5-273](#).

Return to the [Summary Table](#).

**Table 5-273. HW\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

### 5.2.2.105 MSS\_CR5F\_CLK\_SRC\_SEL\_CTRL Register (Offset = 1DCh) [Reset = 00000000h]

MSS\_CR5F\_CLK\_SRC\_SEL\_CTRL is shown in [Table 5-274](#).

Return to the [Summary Table](#).

**Table 5-274. MSS\_CR5F\_CLK\_SRC\_SEL\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	clksrcsel	R/W	0h	writing 3'b111 ensures R5 to be same as BUS_CLK writing 3'b000 ensures R5 clock will be same as CR5_CLK from top_rcm

### 5.2.2.106 MSS\_CPSW\_MII\_CLK\_SRC\_SEL Register (Offset = 1E0h) [Reset = 00000000h]

MSS\_CPSW\_MII\_CLK\_SRC\_SEL is shown in [Table 5-275](#).

Return to the [Summary Table](#).

**Table 5-275. MSS\_CPSW\_MII\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for CPSW_MII. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.107 MSS\_CPSW\_MII\_CLK\_STATUS Register (Offset = 1E4h) [Reset = 00000001h]

MSS\_CPSW\_MII\_CLK\_STATUS is shown in [Table 5-276](#).

Return to the [Summary Table](#).

**Table 5-276. MSS\_CPSW\_MII\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	clkinuse	R	1h	Status shows the source clock selected for CPSW_MII

### 5.2.2.108 MSS\_L2\_BANKC\_PD\_CTRL Register (Offset = 1E8h) [Reset = 000007XXh]

MSS\_L2\_BANKC\_PD\_CTRL is shown in [Table 5-277](#).

Return to the [Summary Table](#).

**Table 5-277. MSS\_L2\_BANKC\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW control for power signal 'AGOODIN' for MSS_L2_BANKC
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW control for power signal 'AONIN' for MSS_L2_BANKC
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW control for power signal 'ISO' for MSS_L2_BANKC

### 5.2.2.109 MSS\_L2\_BANKC\_PD\_STATUS Register (Offset = 1ECh) [Reset = 00000003h]

MSS\_L2\_BANKC\_PD\_STATUS is shown in [Table 5-278](#).

Return to the [Summary Table](#).

**Table 5-278. MSS\_L2\_BANKC\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	agoodout	R	1h	SW status indicating the 'pgoodin' of MSS_L2_BANKC
0	aonout	R	1h	SW status indicating the 'ponin' of MSS_L2_BANKC



### 5.2.2.110 MSS\_IP\_CLK\_CFG Register (Offset = 1F0h) [Reset = 00000000h]

MSS\_IP\_CLK\_CFG is shown in [Table 5-279](#).

Return to the [Summary Table](#).

**Table 5-279. MSS\_IP\_CLK\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	gate	R/W	0h	IP clock gating configuration bits. Data should be loaded as multibit. Writing 3'b111 will gate the IP and disable the slave

### 5.2.2.111 HSM\_RTIA\_CLK\_SRC\_SEL Register (Offset = 400h) [Reset = 00000000h]

HSM\_RTIA\_CLK\_SRC\_SEL is shown in [Table 5-280](#).

Return to the [Summary Table](#).

**Table 5-280. HSM\_RTIA\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for HSM_RTIA. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.112 HSM\_WDT\_CLK\_SRC\_SEL Register (Offset = 404h) [Reset = 00000000h]

HSM\_WDT\_CLK\_SRC\_SEL is shown in [Table 5-281](#).

Return to the [Summary Table](#).

**Table 5-281. HSM\_WDT\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for HSM_WDT. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.113 HSM\_RTC\_CLK\_SRC\_SEL Register (Offset = 408h) [Reset = 00000777h]

HSM\_RTC\_CLK\_SRC\_SEL is shown in [Table 5-282](#).

Return to the [Summary Table](#).

**Table 5-282. HSM\_RTC\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	777h	Select line for selecting source clock for HSM_RTC. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.114 HSM\_DMTA\_CLK\_SRC\_SEL Register (Offset = 40Ch) [Reset = 00000000h]

HSM\_DMTA\_CLK\_SRC\_SEL is shown in [Table 5-283](#).

Return to the [Summary Table](#).

**Table 5-283. HSM\_DMTA\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for HSM_DMTA. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.115 HSM\_DMTB\_CLK\_SRC\_SEL Register (Offset = 410h) [Reset = 00000000h]

HSM\_DMTB\_CLK\_SRC\_SEL is shown in [Table 5-284](#).

Return to the [Summary Table](#).

**Table 5-284. HSM\_DMTB\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for HSM_DMTB. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

### 5.2.2.116 HSM\_RTI\_CLK\_DIV\_VAL Register (Offset = 414h) [Reset = 00000000h]

HSM\_RTI\_CLK\_DIV\_VAL is shown in [Table 5-285](#).

Return to the [Summary Table](#).

**Table 5-285. HSM\_RTI\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value HSM RTI selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.117 HSM\_WDT\_CLK\_DIV\_VAL Register (Offset = 418h) [Reset = 00000000h]

HSM\_WDT\_CLK\_DIV\_VAL is shown in [Table 5-286](#).

Return to the [Summary Table](#).

**Table 5-286. HSM\_WDT\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value HSM WDT selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference



### 5.2.2.118 HSM\_RTC\_CLK\_DIV\_VAL Register (Offset = 41Ch) [Reset = 00000000h]

HSM\_RTC\_CLK\_DIV\_VAL is shown in [Table 5-287](#).

Return to the [Summary Table](#).

**Table 5-287. HSM\_RTC\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value HSM RTC selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.119 HSM\_DMTA\_CLK\_DIV\_VAL Register (Offset = 420h) [Reset = 00000000h]

HSM\_DMTA\_CLK\_DIV\_VAL is shown in [Table 5-288](#).

Return to the [Summary Table](#).

**Table 5-288. HSM\_DMTA\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value HSM DMTA selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.120 HSM\_DMTB\_CLK\_DIV\_VAL Register (Offset = 424h) [Reset = 00000000h]

HSM\_DMTB\_CLK\_DIV\_VAL is shown in [Table 5-289](#).

Return to the [Summary Table](#).

**Table 5-289. HSM\_DMTB\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value HSM DMTB selected clock. Data should be loaded as multibit. For example: if divider value of '0x8' should be selected then '0x888' should be configured to the register. Refer to TPR12 clock spec for clock reference

### 5.2.2.121 HSM\_RTI\_CLK\_GATE Register (Offset = 428h) [Reset = 00000000h]

HSM\_RTI\_CLK\_GATE is shown in [Table 5-290](#).

Return to the [Summary Table](#).

**Table 5-290. HSM\_RTI\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for HSM RTI

### 5.2.2.122 HSM\_WDT\_CLK\_GATE Register (Offset = 42Ch) [Reset = 00000000h]

HSM\_WDT\_CLK\_GATE is shown in [Table 5-291](#).

Return to the [Summary Table](#).

**Table 5-291. HSM\_WDT\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for HSM WDT

### 5.2.2.123 HSM\_RTC\_CLK\_GATE Register (Offset = 430h) [Reset = 00000000h]

HSM\_RTC\_CLK\_GATE is shown in [Table 5-292](#).

Return to the [Summary Table](#).

**Table 5-292. HSM\_RTC\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for HSM RTC

### 5.2.2.124 HSM\_DMTA\_CLK\_GATE Register (Offset = 434h) [Reset = 00000000h]

HSM\_DMTA\_CLK\_GATE is shown in [Table 5-293](#).

Return to the [Summary Table](#).

**Table 5-293. HSM\_DMTA\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for HSM DMTA

### 5.2.2.125 HSM\_DMTB\_CLK\_GATE Register (Offset = 438h) [Reset = 00000000h]

HSM\_DMTB\_CLK\_GATE is shown in [Table 5-294](#).

Return to the [Summary Table](#).

**Table 5-294. HSM\_DMTB\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing '111' will gate clock for HSM DMTB



### 5.2.2.126 HSM\_RTI\_CLK\_STATUS Register (Offset = 43Ch) [Reset = 00000001h]

HSM\_RTI\_CLK\_STATUS is shown in [Table 5-295](#).

Return to the [Summary Table](#).

**Table 5-295. HSM\_RTI\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for HSM_RTI
7-0	clkinuse	R	1h	Status shows the source clock slected for HSM_RTI

### 5.2.2.127 HSM\_WDT\_CLK\_STATUS Register (Offset = 440h) [Reset = 00000001h]

HSM\_WDT\_CLK\_STATUS is shown in [Table 5-296](#).

Return to the [Summary Table](#).

**Table 5-296. HSM\_WDT\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for HSM_WDT
7-0	clkinuse	R	1h	Status shows the source clock slected for HSM_WDT

### 5.2.2.128 HSM\_RTC\_CLK\_STATUS Register (Offset = 444h) [Reset = 00000080h]

HSM\_RTC\_CLK\_STATUS is shown in [Table 5-297](#).

Return to the [Summary Table](#).

**Table 5-297. HSM\_RTC\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for HSM_RTC
7-0	clkinuse	R	80h	Status shows the source clock slected for HSM_RTC

### 5.2.2.129 HSM\_DMTA\_CLK\_STATUS Register (Offset = 448h) [Reset = 00000001h]

HSM\_DMTA\_CLK\_STATUS is shown in [Table 5-298](#).

Return to the [Summary Table](#).

**Table 5-298. HSM\_DMTA\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for HSM_DMTA
7-0	clkinuse	R	1h	Status shows the source clock slected for HSM_DMTA

### 5.2.2.130 HSM\_DMTB\_CLK\_STATUS Register (Offset = 44Ch) [Reset = 00000001h]

HSM\_DMTB\_CLK\_STATUS is shown in [Table 5-299](#).

Return to the [Summary Table](#).

**Table 5-299. HSM\_DMTB\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for HSM_DMTB
7-0	clkinuse	R	1h	Status shows the source clock slected for HSM_DMTB

### 5.2.2.131 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-300](#).

Return to the [Summary Table](#).

**Table 5-300. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.2.132 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-301](#).

Return to the [Summary Table](#).

**Table 5-301. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.2.133 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-302](#).

Return to the [Summary Table](#).

**Table 5-302. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D



### 5.2.2.134 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-303](#).

Return to the [Summary Table](#).

**Table 5-303. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.2.135 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-304](#).

Return to the [Summary Table](#).

**Table 5-304. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.2.136 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-305](#).

Return to the [Summary Table](#).

**Table 5-305. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

### 5.2.2.137 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-306](#).

Return to the [Summary Table](#).

**Table 5-306. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

### 5.2.2.138 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-307](#).

Return to the [Summary Table](#).

**Table 5-307. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.2.139 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-308](#).

Return to the [Summary Table](#).

**Table 5-308. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

### 5.2.2.140 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-309](#).

Return to the [Summary Table](#).

**Table 5-309. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.2.141 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-310](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-310. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component



### 5.2.2.142 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-311](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-311. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.2.143 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-312](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-312. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.2.144 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-313](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-313. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.2.145 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-314](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-314. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.2.146 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-315](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-315. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.2.147 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-316](#).

Return to the [Summary Table](#).

EOI register

**Table 5-316. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.2.148 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-317](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-317. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.2.149 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-318](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-318. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault



### 5.2.2.150 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-319](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-319. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.2.151 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-320](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-320. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

### 5.2.3 DSS\_RCM Registers

Table 5-321 lists the memory-mapped registers for the DSS\_RCM registers. All register offset addresses not listed in Table 5-321 should be considered as reserved locations and the register contents should not be modified.

**Table 5-321. DSS\_RCM Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.3.1</a>
4h	HW_REG0		<a href="#">Section 5.2.3.2</a>
8h	HW_REG1		<a href="#">Section 5.2.3.3</a>
Ch	PREVIOUS_NAME		<a href="#">Section 5.2.3.4</a>
10h	HW_REG3		<a href="#">Section 5.2.3.5</a>
90h	DSS_HWA_CLK_SRC_SEL		<a href="#">Section 5.2.3.6</a>
BCh	DSS_HWA_CLK_GATE		<a href="#">Section 5.2.3.7</a>
D0h	DSS_CBUFF_CLK_GATE		<a href="#">Section 5.2.3.8</a>
D8h	DSS_HWA_CLK_STATUS		<a href="#">Section 5.2.3.9</a>
10Ch	DSS_MCRC_RST_CTRL		<a href="#">Section 5.2.3.10</a>
118h	DSS_L3_BANKA0_PD_CTRL		<a href="#">Section 5.2.3.11</a>
11Ch	DSS_L3_BANKA1_PD_CTRL		<a href="#">Section 5.2.3.12</a>
128h	DSS_L3_BANKB0_PD_CTRL		<a href="#">Section 5.2.3.13</a>
12Ch	DSS_L3_BANKB1_PD_CTRL		<a href="#">Section 5.2.3.14</a>
158h	DSS_HWA_PD_CTRL		<a href="#">Section 5.2.3.15</a>
160h	DSS_L3_BANKA0_PD_STATUS		<a href="#">Section 5.2.3.16</a>
164h	DSS_L3_BANKA1_PD_STATUS		<a href="#">Section 5.2.3.17</a>
170h	DSS_L3_BANKB0_PD_STATUS		<a href="#">Section 5.2.3.18</a>
174h	DSS_L3_BANKB1_PD_STATUS		<a href="#">Section 5.2.3.19</a>
1A0h	DSS_HWA_PD_STATUS		<a href="#">Section 5.2.3.20</a>
1B0h	DSS_L3_PD_CTRL_STICKYBIT		<a href="#">Section 5.2.3.21</a>
1C8h	DSS_HWA_RST_CTRL		<a href="#">Section 5.2.3.22</a>
1CCh	DSS_HWA_RST_CTRL		<a href="#">Section 5.2.3.22</a>
1D8h	DSS_EDMA_RST_CTRL		<a href="#">Section 5.2.3.23</a>
1E4h	DSS_CSI_CLK_GATE		<a href="#">Section 5.2.3.24</a>
1E8h	DSS_IP_CLK_CFG		<a href="#">Section 5.2.3.25</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.3.26</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.3.27</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.3.28</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.3.29</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.3.30</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.3.31</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.3.32</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.3.33</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.3.34</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.3.35</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.3.36</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.3.37</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.3.38</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.3.39</a>

**Table 5-321. DSS\_RCM Registers (continued)**

Offset	Acronym	Register Name	Section
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.3.40</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.3.41</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.3.42</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.3.43</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.3.44</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.3.45</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.3.46</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-322](#) shows the codes that are used for access types in this section.

**Table 5-322. DSS\_RCM Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.3.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-323](#).

Return to the [Summary Table](#).

PID register

**Table 5-323. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.3.2 HW\_REG0 Register (Offset = 4h) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-324](#).

Return to the [Summary Table](#).

**Table 5-324. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved register

### 5.2.3.3 HW\_REG1 Register (Offset = 8h) [Reset = 00000000h]

HW\_REG1 is shown in [Table 5-325](#).

Return to the [Summary Table](#).

**Table 5-325. HW\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

### 5.2.3.4 PREVIOUS\_NAME Register (Offset = Ch) [Reset = 00000000h]

PREVIOUS\_NAME is shown in [Table 5-326](#).

Return to the [Summary Table](#).

**Table 5-326. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser



### 5.2.3.5 HW\_REG3 Register (Offset = 10h) [Reset = 00000000h]

HW\_REG3 is shown in [Table 5-327](#).

Return to the [Summary Table](#).

**Table 5-327. HW\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

### 5.2.3.6 DSS\_HWA\_CLK\_SRC\_SEL Register (Offset = 90h) [Reset = 00000000h]

DSS\_HWA\_CLK\_SRC\_SEL is shown in [Table 5-328](#).

Return to the [Summary Table](#).

**Table 5-328. DSS\_HWA\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	clksrcsel	R/W	0h	Select line for selecting source clock for DSS HWA. Data should be loaded as multibit. Write 3'b000 : TOPRCM_CR5_CLK Write 3'b111 : TOPRCM_SYS_CLK

### 5.2.3.7 DSS\_HWA\_CLK\_GATE Register (Offset = BCh) [Reset = 00000000h]

DSS\_HWA\_CLK\_GATE is shown in [Table 5-329](#).

Return to the [Summary Table](#).

**Table 5-329. DSS\_HWA\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for DSS HWA Data should be loaded as multibit. Write 3'b000 : Clock is ungated Write 3'b111 : Clock is gated

### 5.2.3.8 DSS\_CBUFF\_CLK\_GATE Register (Offset = D0h) [Reset = 00000000h]

DSS\_CBUFF\_CLK\_GATE is shown in [Table 5-330](#).

Return to the [Summary Table](#).

**Table 5-330. DSS\_CBUFF\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for DSS CBUFF Data should be loaded as multibit. Write 3'b000 : Clock is ungated Write 3'b111 : Clock is gated

### 5.2.3.9 DSS\_HWA\_CLK\_STATUS Register (Offset = D8h) [Reset = 00000001h]

DSS\_HWA\_CLK\_STATUS is shown in [Table 5-331](#).

Return to the [Summary Table](#).

**Table 5-331. DSS\_HWA\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	clkinuse	R	1h	Status shows the source clock selected for DSS HWA Clock

### 5.2.3.10 DSS\_MCRC\_RST\_CTRL Register (Offset = 10Ch) [Reset = 00000000h]

DSS\_MCRC\_RST\_CTRL is shown in [Table 5-332](#).

Return to the [Summary Table](#).

**Table 5-332. DSS\_MCRC\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug purpose only. Software needs to ensure the state of Device/IP before configuring. Writing 3'b11 will assert reset for DSS MCRC

### 5.2.3.11 DSS\_L3\_BANKA0\_PD\_CTRL Register (Offset = 118h) [Reset = 000007XXh]

DSS\_L3\_BANKA0\_PD\_CTRL is shown in [Table 5-333](#).

Return to the [Summary Table](#).

**Table 5-333. DSS\_L3\_BANKA0\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL1
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL0
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW Control for <IP>_PD_CTRL Isolation

### 5.2.3.12 DSS\_L3\_BANKA1\_PD\_CTRL Register (Offset = 11Ch) [Reset = 000007XXh]

DSS\_L3\_BANKA1\_PD\_CTRL is shown in [Table 5-334](#).

Return to the [Summary Table](#).

**Table 5-334. DSS\_L3\_BANKA1\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL1
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL0
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW Control for <IP>_PD_CTRL Isolation



### 5.2.3.13 DSS\_L3\_BANKB0\_PD\_CTRL Register (Offset = 128h) [Reset = 000007XXh]

DSS\_L3\_BANKB0\_PD\_CTRL is shown in [Table 5-335](#).

Return to the [Summary Table](#).

**Table 5-335. DSS\_L3\_BANKB0\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL1
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL0
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW Control for <IP>_PD_CTRL Isolation

### 5.2.3.14 DSS\_L3\_BANKB1\_PD\_CTRL Register (Offset = 12Ch) [Reset = 000007XXh]

DSS\_L3\_BANKB1\_PD\_CTRL is shown in [Table 5-336](#).

Return to the [Summary Table](#).

**Table 5-336. DSS\_L3\_BANKB1\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL1
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL0
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW Control for <IP>_PD_CTRL Isolation

### 5.2.3.15 DSS\_HWA\_PD\_CTRL Register (Offset = 158h) [Reset = 0007XXXXh]

DSS\_HWA\_PD\_CTRL is shown in [Table 5-337](#).

Return to the [Summary Table](#).

**Table 5-337. DSS\_HWA\_PD\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	pgoodin	R/W	7h	SW Control for <IP>_PD_CTRL Power up CTRL1
15	RESERVED	R/W	0h	
14-12	ponin	R/W	7h	SW Control for <IP>_PD_CTRL Power up CTRL0
11	RESERVED	R/W	0h	
10-8	agoodin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL1
7	RESERVED	R/W	0h	
6-4	aonin	R/W	7h	SW Control for <IP>_PD_CTRL Memory Array Power up CTRL0
3	RESERVED	R/W	0h	
2-0	iso	R/W	0h	SW Control for <IP>_PD_CTRL Isolation

### 5.2.3.16 DSS\_L3\_BANKA0\_PD\_STATUS Register (Offset = 160h) [Reset = 000000Fh]

DSS\_L3\_BANKA0\_PD\_STATUS is shown in [Table 5-338](#).

Return to the [Summary Table](#).

**Table 5-338. DSS\_L3\_BANKA0\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	agoodin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL1
2	aonin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL0
1	agoodout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL1
0	aonout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL0

### 5.2.3.17 DSS\_L3\_BANKA1\_PD\_STATUS Register (Offset = 164h) [Reset = 000000Fh]

DSS\_L3\_BANKA1\_PD\_STATUS is shown in [Table 5-339](#).

Return to the [Summary Table](#).

**Table 5-339. DSS\_L3\_BANKA1\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	agoodin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL1
2	aonin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL0
1	agoodout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL1
0	aonout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL0

### 5.2.3.18 DSS\_L3\_BANKB0\_PD\_STATUS Register (Offset = 170h) [Reset = 000000Fh]

DSS\_L3\_BANKB0\_PD\_STATUS is shown in [Table 5-340](#).

Return to the [Summary Table](#).

**Table 5-340. DSS\_L3\_BANKB0\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	agoodin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL1
2	aonin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL0
1	agoodout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL1
0	aonout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL0

### 5.2.3.19 DSS\_L3\_BANKB1\_PD\_STATUS Register (Offset = 174h) [Reset = 000000Fh]

DSS\_L3\_BANKB1\_PD\_STATUS is shown in [Table 5-341](#).

Return to the [Summary Table](#).

**Table 5-341. DSS\_L3\_BANKB1\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	agoodin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL1
2	aonin	R	1h	Status for sticky control <IP>_PD_CTRL Memory Array Power up CTRL0
1	agoodout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL1
0	aonout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL0

### 5.2.3.20 DSS\_HWA\_PD\_STATUS Register (Offset = 1A0h) [Reset = 0000000Fh]

DSS\_HWA\_PD\_STATUS is shown in [Table 5-342](#).

Return to the [Summary Table](#).

**Table 5-342. DSS\_HWA\_PD\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	pgoodout	R	1h	Status for <IP>_PD_CTRL Power up CTRL1
2	ponout	R	1h	Status for <IP>_PD_CTRL Power up CTRL0
1	agoodout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL1
0	aonout	R	1h	Status for <IP>_PD_CTRL Memory Array Power up CTRL0



### 5.2.3.21 DSS\_L3\_PD\_CTRL\_STICKYBIT Register (Offset = 1B0h) [Reset = 00000000h]

DSS\_L3\_PD\_CTRL\_STICKYBIT is shown in [Table 5-343](#).

Return to the [Summary Table](#).

**Table 5-343. DSS\_L3\_PD\_CTRL\_STICKYBIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	set	R/W	0h	Sticky bit for DSS L3 PD CTRL. Write 3'b111 to lock the configuration of DSS_L3_BANK*_PD_CTRL. Once this field is written, there is no impact of changing the value of aonin and agoodin fields in DSS_L3_BANK*_PD_CTRL registers

### 5.2.3.22 DSS\_HWA\_RST\_CTRL Register (Offset = 1C8h) [Reset = 00000000h]

DSS\_HWA\_RST\_CTRL is shown in [Table 5-344](#).

Return to the [Summary Table](#).

**Table 5-344. DSS\_HWA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This register is for Debug Purposes only. Reset control for DSS HWA Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW

### 5.2.3.23 DSS\_EDMA\_RST\_CTRL Register (Offset = 1D8h) [Reset = 0000000Xh]

DSS\_EDMA\_RST\_CTRL is shown in [Table 5-345](#).

Return to the [Summary Table](#).

**Table 5-345. DSS\_EDMA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6-4	assert_tc1	R/W	0h	This register is for Debug Purposes only. Reset control for DSS TPTCA1 Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW
3	RESERVED	R/W	0h	
2-0	assert_tc0	R/W	0h	This register is for Debug Purposes only. Reset control for DSS TPTCA0 Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW

### 5.2.3.24 DSS\_CSI\_CLK\_GATE Register (Offset = 1E4h) [Reset = 00000000h]

DSS\_CSI\_CLK\_GATE is shown in [Table 5-346](#).

Return to the [Summary Table](#).

**Table 5-346. DSS\_CSI\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for DSS CSI2-TX Data should be loaded as multibit. Write 3'b000 : Clock is ungated Write 3'b111 : Clock is gated

### 5.2.3.25 DSS\_IP\_CLK\_CFG Register (Offset = 1E8h) [Reset = 00000000h]

DSS\_IP\_CLK\_CFG is shown in [Table 5-347](#).

Return to the [Summary Table](#).

**Table 5-347. DSS\_IP\_CLK\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	gate	R/W	0h	IP clock gating configuration bits. Data should be loaded as multibit. Writing 3'b111 will gate the IP and disable the slave 2:0 - MCRC

### 5.2.3.26 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-348](#).

Return to the [Summary Table](#).

**Table 5-348. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.3.27 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-349](#).

Return to the [Summary Table](#).

**Table 5-349. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.3.28 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-350](#).

Return to the [Summary Table](#).

**Table 5-350. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D



### 5.2.3.29 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-351](#).

Return to the [Summary Table](#).

**Table 5-351. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.3.30 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-352](#).

Return to the [Summary Table](#).

**Table 5-352. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.3.31 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-353](#).

Return to the [Summary Table](#).

**Table 5-353. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

### 5.2.3.32 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-354](#).

Return to the [Summary Table](#).

**Table 5-354. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

### 5.2.3.33 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-355](#).

Return to the [Summary Table](#).

**Table 5-355. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.3.34 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-356](#).

Return to the [Summary Table](#).

**Table 5-356. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

### 5.2.3.35 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-357](#).

Return to the [Summary Table](#).

**Table 5-357. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.3.36 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-358](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-358. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component



### 5.2.3.37 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-359](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-359. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.3.38 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-360](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-360. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.3.39 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-361](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-361. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.3.40 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-362](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-362. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.3.41 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-363](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-363. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.3.42 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-364](#).

Return to the [Summary Table](#).

EOI register

**Table 5-364. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.3.43 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-365](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-365. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.3.44 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-366](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-366. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault



### 5.2.3.45 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-367](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-367. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.3.46 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-368](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-368. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

### 5.2.4 RSS\_RCM Registers

Table 5-369 lists the memory-mapped registers for the RSS\_RCM registers. All register offset addresses not listed in Table 5-369 should be considered as reserved locations and the register contents should not be modified.

**Table 5-369. RSS\_RCM Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.4.1</a>
4h	HW_REG0		<a href="#">Section 5.2.4.2</a>
8h	HW_REG1		<a href="#">Section 5.2.4.3</a>
Ch	PREVIOUS_NAME		<a href="#">Section 5.2.4.4</a>
10h	HW_REG3		<a href="#">Section 5.2.4.5</a>
18h	RSS_BSS_SYS_CLK_GATE		<a href="#">Section 5.2.4.6</a>
20h	RSS_EDMA_RST_CTRL		<a href="#">Section 5.2.4.7</a>
24h	RSS_BSS_RST_CTRL		<a href="#">Section 5.2.4.8</a>
28h	RSS_FRC_CLK_SRC_SEL		<a href="#">Section 5.2.4.9</a>
2Ch	RSS_FRC_CLK_GATE		<a href="#">Section 5.2.4.10</a>
30h	RSS_FRC_CLK_DIV_VAL		<a href="#">Section 5.2.4.11</a>
34h	RSS_FRC_CLK_STATUS		<a href="#">Section 5.2.4.12</a>
38h	RSS_FRC_RST_CTRL		<a href="#">Section 5.2.4.13</a>
3Ch	RSS_IP_CLK_CFG		<a href="#">Section 5.2.4.14</a>
40h	RSS_IP_ACCESS_DIS		<a href="#">Section 5.2.4.15</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.4.16</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.4.17</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.4.18</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.4.19</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.4.20</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.4.21</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.4.22</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.4.23</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.4.24</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.4.25</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.4.26</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.4.27</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.4.28</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.4.29</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.4.30</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.4.31</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.4.32</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.4.33</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.4.34</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.4.35</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.4.36</a>

Complex bit access types are encoded to fit into small table cells. Table 5-370 shows the codes that are used for access types in this section.

**Table 5-370. RSS\_RCM Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.4.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-371](#).

Return to the [Summary Table](#).

PID register

**Table 5-371. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.4.2 HW\_REG0 Register (Offset = 4h) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-372](#).

Return to the [Summary Table](#).

**Table 5-372. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved register

### 5.2.4.3 HW\_REG1 Register (Offset = 8h) [Reset = 00000000h]

HW\_REG1 is shown in [Table 5-373](#).

Return to the [Summary Table](#).

**Table 5-373. HW\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

#### 5.2.4.4 PREVIOUS\_NAME Register (Offset = Ch) [Reset = 00000000h]

PREVIOUS\_NAME is shown in [Table 5-374](#).

Return to the [Summary Table](#).

**Table 5-374. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser



#### 5.2.4.5 HW\_REG3 Register (Offset = 10h) [Reset = 00000000h]

HW\_REG3 is shown in [Table 5-375](#).

Return to the [Summary Table](#).

**Table 5-375. HW\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg	R/W	0h	HW Reserved regiser

#### 5.2.4.6 RSS\_BSS\_SYS\_CLK\_GATE Register (Offset = 18h) [Reset = 00000000h]

RSS\_BSS\_SYS\_CLK\_GATE is shown in [Table 5-376](#).

Return to the [Summary Table](#).

**Table 5-376. RSS\_BSS\_SYS\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	Clock gating config for RSS BSS Data should be loaded as multibit. Write 3'b000 : Clock is ungated Write 3'b111 : Clock is gated

#### 5.2.4.7 RSS\_EDMA\_RST\_CTRL Register (Offset = 20h) [Reset = 000000XXh]

RSS\_EDMA\_RST\_CTRL is shown in [Table 5-377](#).

Return to the [Summary Table](#).

**Table 5-377. RSS\_EDMA\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	tpcca0_assert	R/W	0h	writing '111' will reset MSS_TPCCA
7	RESERVED	R/W	0h	
6-4	tpcca_assert	R/W	0h	This feature is for debug pupose only. software need to ensure the correct state of Device/IP before configuring this reset control for RSS EDMA Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW
3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug pupose only. software need to ensure the correct state of Device/IP before configuring this reset control for RSS EDMA Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW

### 5.2.4.8 RSS\_BSS\_RST\_CTRL Register (Offset = 24h) [Reset = 00000000h]

RSS\_BSS\_RST\_CTRL is shown in [Table 5-378](#).

Return to the [Summary Table](#).

**Table 5-378. RSS\_BSS\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug pupose only. software need to ensure the correct state of Device/IP before configuring this reset control for RSS BSS Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW

#### 5.2.4.9 RSS\_FRC\_CLK\_SRC\_SEL Register (Offset = 28h) [Reset = 00000000h]

RSS\_FRC\_CLK\_SRC\_SEL is shown in [Table 5-379](#).

Return to the [Summary Table](#).

**Table 5-379. RSS\_FRC\_CLK\_SRC\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clksrcsel	R/W	0h	Select line for selecting source clock for FRC. Data should be loaded as multibit. For example: if '0x5' should be selected then '0x555' should be configured to the register. Refer to TPR12 clock spec for source clock reference

#### 5.2.4.10 RSS\_FRC\_CLK\_GATE Register (Offset = 2Ch) [Reset = 00000000h]

RSS\_FRC\_CLK\_GATE is shown in [Table 5-380](#).

Return to the [Summary Table](#).

**Table 5-380. RSS\_FRC\_CLK\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gated	R/W	0h	writing 3'b111 will gate the clock for FRC

#### 5.2.4.11 RSS\_FRC\_CLK\_DIV\_VAL Register (Offset = 30h) [Reset = 00000000h]

RSS\_FRC\_CLK\_DIV\_VAL is shown in [Table 5-381](#).

Return to the [Summary Table](#).

**Table 5-381. RSS\_FRC\_CLK\_DIV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	clkdivr	R/W	0h	Divider value for FRC selected clock. Data should be loaded as multibit. For example: if divider value of 8(1000) should be selected then '100010001000' should be configured to the register. Refer to TPR12 clock planner for clock reference

#### 5.2.4.12 RSS\_FRC\_CLK\_STATUS Register (Offset = 34h) [Reset = 00000001h]

RSS\_FRC\_CLK\_STATUS is shown in [Table 5-382](#).

Return to the [Summary Table](#).

**Table 5-382. RSS\_FRC\_CLK\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	currdivider	R	0h	Status shows the current divider value choosen for FRC
7-0	clkinuse	R	1h	Status shows the source clock slected for FRC



#### 5.2.4.13 RSS\_FRC\_RST\_CTRL Register (Offset = 38h) [Reset = 00000000h]

RSS\_FRC\_RST\_CTRL is shown in [Table 5-383](#).

Return to the [Summary Table](#).

**Table 5-383. RSS\_FRC\_RST\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	assert	R/W	0h	This feature is for debug pupose only. software need to ensure the correct state of Device/IP before configuring this reset control for RSS FRC Data should be loaded as multibit. Write 3'b000: Reset is not asserted by SW. There could be another reset source which could reset the module. Refer to RCM spec for more details Write 3'b111 : Reset is asserted by SW

#### 5.2.4.14 RSS\_IP\_CLK\_CFG Register (Offset = 3Ch) [Reset = 00000000h]

RSS\_IP\_CLK\_CFG is shown in [Table 5-384](#).

Return to the [Summary Table](#).

**Table 5-384. RSS\_IP\_CLK\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	gate	R/W	0h	IP clock gating configuration bits Writing 3'b111 will gate the IP

#### 5.2.4.15 RSS\_IP\_ACCESS\_DIS Register (Offset = 40h) [Reset = 00000000h]

RSS\_IP\_ACCESS\_DIS is shown in [Table 5-385](#).

Return to the [Summary Table](#).

**Table 5-385. RSS\_IP\_ACCESS\_DIS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	disable	R/W	0h	IP access disable configuration bits Writing 3'b111 will disable the IP

#### 5.2.4.16 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-386](#).

Return to the [Summary Table](#).

**Table 5-386. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

#### 5.2.4.17 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-387](#).

Return to the [Summary Table](#).

**Table 5-387. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

#### 5.2.4.18 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-388](#).

Return to the [Summary Table](#).

**Table 5-388. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D

#### 5.2.4.19 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-389](#).

Return to the [Summary Table](#).

**Table 5-389. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

#### 5.2.4.20 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-390](#).

Return to the [Summary Table](#).

**Table 5-390. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D



#### 5.2.4.21 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-391](#).

Return to the [Summary Table](#).

**Table 5-391. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

#### 5.2.4.22 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-392](#).

Return to the [Summary Table](#).

**Table 5-392. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

#### 5.2.4.23 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-393](#).

Return to the [Summary Table](#).

**Table 5-393. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

#### 5.2.4.24 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-394](#).

Return to the [Summary Table](#).

**Table 5-394. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

#### 5.2.4.25 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-395](#).

Return to the [Summary Table](#).

**Table 5-395. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

#### 5.2.4.26 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-396](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-396. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

#### 5.2.4.27 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-397](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-397. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.4.28 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-398](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-398. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.



#### 5.2.4.29 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-399](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-399. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.4.30 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-400](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-400. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.4.31 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-401](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-401. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.4.32 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-402](#).

Return to the [Summary Table](#).

EOI register

**Table 5-402. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.4.33 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-403](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-403. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.4.34 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-404](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-404. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault

### 5.2.4.35 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-405](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-405. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.4.36 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-406](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-406. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.



### 5.2.5 DSS\_CTRL Registers

Table 5-407 lists the memory-mapped registers for the DSS\_CTRL registers. All register offset addresses not listed in Table 5-407 should be considered as reserved locations and the register contents should not be modified.

**Table 5-407. DSS\_CTRL Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.5.1</a>
4h	HW_REG0		<a href="#">Section 5.2.5.2</a>
8h	HW_REG1		<a href="#">Section 5.2.5.3</a>
Ch	PREVIOUS_NAME		<a href="#">Section 5.2.5.4</a>
10h	HW_REG3		<a href="#">Section 5.2.5.5</a>
14h	DSS_SW_INT		<a href="#">Section 5.2.5.6</a>
18h	DSS_TPCC_A_ERRAGG_MASK		<a href="#">Section 5.2.5.7</a>
1Ch	DSS_TPCC_A_ERRAGG_STATUS		<a href="#">Section 5.2.5.8</a>
20h	DSS_TPCC_A_ERRAGG_STATUS_RAW		<a href="#">Section 5.2.5.9</a>
24h	DSS_TPCC_A_INTAGG_MASK		<a href="#">Section 5.2.5.10</a>
28h	DSS_TPCC_A_INTAGG_STATUS		<a href="#">Section 5.2.5.11</a>
2Ch	DSS_TPCC_A_INTAGG_STATUS_RAW		<a href="#">Section 5.2.5.12</a>
60h	DSS_TPCC_MEMINIT_START		<a href="#">Section 5.2.5.13</a>
64h	DSS_TPCC_MEMINIT_STATUS		<a href="#">Section 5.2.5.14</a>
68h	DSS_TPCC_MEMINIT_DONE		<a href="#">Section 5.2.5.15</a>
98h	DSS_L3RAM_MEMINIT_START		<a href="#">Section 5.2.5.16</a>
9Ch	DSS_L3RAM_MEMINIT_STATUS		<a href="#">Section 5.2.5.17</a>
A0h	DSS_L3RAM_MEMINIT_DONE		<a href="#">Section 5.2.5.18</a>
BCh	DSS_TPCC_A_PARITY_CTRL		<a href="#">Section 5.2.5.19</a>
C8h	DSS_TPCC_A_PARITY_STATUS		<a href="#">Section 5.2.5.20</a>
D4h	TPTC_DBS_CONFIG		<a href="#">Section 5.2.5.21</a>
D8h	DSS_DSP_BOOTCFG		<a href="#">Section 5.2.5.22</a>
DCh	DSS_DSP_NMI_GATE		<a href="#">Section 5.2.5.23</a>
E0h	DSS_PBIIST_KEY_RESET		<a href="#">Section 5.2.5.24</a>
E4h	DSS_PBIIST_REG0		<a href="#">Section 5.2.5.25</a>
E8h	DSS_PBIIST_REG1		<a href="#">Section 5.2.5.26</a>
ECh	DSS_TPTC_BOUNDARY_CFG0		<a href="#">Section 5.2.5.27</a>
F0h	DSS_TPTC_BOUNDARY_CFG1		<a href="#">Section 5.2.5.28</a>
F4h	DSS_TPTC_BOUNDARY_CFG2		<a href="#">Section 5.2.5.29</a>
F8h	DSS_TPTC_XID_REORDER_CFG0		<a href="#">Section 5.2.5.30</a>
FCh	DSS_TPTC_XID_REORDER_CFG1		<a href="#">Section 5.2.5.31</a>
100h	DSS_TPTC_XID_REORDER_CFG2		<a href="#">Section 5.2.5.32</a>
108h	ESM_GATING0		<a href="#">Section 5.2.5.33</a>
10Ch	ESM_GATING1		<a href="#">Section 5.2.5.34</a>
110h	ESM_GATING2		<a href="#">Section 5.2.5.35</a>
114h	ESM_GATING3		<a href="#">Section 5.2.5.36</a>
560h	DSS_PERIPH_ERRAGG_MASK0		<a href="#">Section 5.2.5.37</a>
564h	DSS_PERIPH_ERRAGG_STATUS0		<a href="#">Section 5.2.5.38</a>
568h	DSS_PERIPH_ERRAGG_STATUS_RAW0		<a href="#">Section 5.2.5.39</a>
584h	DBG_ACK_CPU_CTRL		<a href="#">Section 5.2.5.40</a>

**Table 5-407. DSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
588h	DBG_ACK_CTL0		<a href="#">Section 5.2.5.41</a>
58Ch	DBG_ACK_CTL1		<a href="#">Section 5.2.5.42</a>
590h	DSS_DSP_INT_SEL		<a href="#">Section 5.2.5.43</a>
594h	DSS_CBUFF_TRIGGER_SEL		<a href="#">Section 5.2.5.44</a>
CE8h	CSI2_TX_PARSTATCFG		<a href="#">Section 5.2.5.45</a>
CECh	CSI2_TX_CFG1		<a href="#">Section 5.2.5.46</a>
CF0h	CSI2_TX_CFG2		<a href="#">Section 5.2.5.47</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.5.48</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.5.49</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.5.50</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.5.51</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.5.52</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.5.53</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.5.54</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.5.55</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.5.56</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.5.57</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.5.58</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.5.59</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.5.60</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.5.61</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.5.62</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.5.63</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.5.64</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.5.65</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.5.66</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.5.67</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-408](#) shows the codes that are used for access types in this section.

**Table 5-408. DSS\_CTRL Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.5.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-409](#).

Return to the [Summary Table](#).

PID register

**Table 5-409. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.5.2 HW\_REG0 Register (Offset = 4h) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-410](#).

Return to the [Summary Table](#).

**Table 5-410. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg0	R/W	0h	Reserved for R&D

### 5.2.5.3 HW\_REG1 Register (Offset = 8h) [Reset = 00000000h]

HW\_REG1 is shown in [Table 5-411](#).

Return to the [Summary Table](#).

**Table 5-411. HW\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg1	R/W	0h	Reserved for R&D

#### 5.2.5.4 PREVIOUS\_NAME Register (Offset = Ch) [Reset = 00000000h]

PREVIOUS\_NAME is shown in [Table 5-412](#).

Return to the [Summary Table](#).

**Table 5-412. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg2	R/W	0h	Reserved for R&D

### 5.2.5.5 HW\_REG3 Register (Offset = 10h) [Reset = 00000000h]

HW\_REG3 is shown in [Table 5-413](#).

Return to the [Summary Table](#).

**Table 5-413. HW\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg3	R/W	0h	Reserved for R&D

### 5.2.5.6 DSS\_SW\_INT Register (Offset = 14h) [Reset = 00000000h]

DSS\_SW\_INT is shown in [Table 5-414](#).

Return to the [Summary Table](#).

**Table 5-414. DSS\_SW\_INT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	dss_swint	R/W	0h	Write pulse bit field: DSS SW Interrupt Write 1 : Generate an interrupt on DSS_SW_INT0



### 5.2.5.7 DSS\_TPCC\_A\_ERRAGG\_MASK Register (Offset = 18h) [Reset = 00XXXXX0h]

DSS\_TPCC\_A\_ERRAGG\_MASK is shown in [Table 5-415](#).

Return to the [Summary Table](#).

**Table 5-415. DSS\_TPCC\_A\_ERRAGG\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	tptc_a1_read_access_error	R/W	0h	Mask Interrupt from DSS_TPTC_A1 to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
25	tptc_a0_read_access_error	R/W	0h	Mask Interrupt from DSS_TPTC_A0 to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
24	tpcc_a_read_access_error	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
23-19	RESERVED	R/W	0h	
18	tptc_a1_write_access_error	R/W	0h	Mask Interrupt from DSS_TPTC_A1 to aggregated Interrupt DSS_TPTC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
17	tptc_a0_write_access_error	R/W	0h	Mask Interrupt from DSS_TPTC_A0 to aggregated Interrupt DSS_TPTC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	tpcc_a_write_access_error	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-9	RESERVED	R/W	0h	
8	tpcc_a_parity_err	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7-4	RESERVED	R/W	0h	
3	tptc_a1_err	R/W	0h	Mask Interrupt from DSS_TPTC_A1 to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	tptc_a0_err	R/W	0h	Mask Interrupt from DSS_TPTC_A0 to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	tpcc_a_mpint	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	tpcc_a_errint	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

### 5.2.5.8 DSS\_TPCC\_A\_ERRAGG\_STATUS Register (Offset = 1Ch) [Reset = 00XXXX0h]

DSS\_TPCC\_A\_ERRAGG\_STATUS is shown in [Table 5-416](#).

Return to the [Summary Table](#).

**Table 5-416. DSS\_TPCC\_A\_ERRAGG\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	tptc_a1_read_access_error	R/W	0h	Status of Interrupt from DSS_TPTC_A1. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
25	tptc_a0_read_access_error	R/W	0h	Status of Interrupt from DSS_TPTC_A0. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
24	tpcc_a_read_access_error	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
23-19	RESERVED	R/W	0h	
18	tptc_a1_write_access_error	R/W	0h	Status of Interrupt from DSS_TPTC_A1. Set only if Interrupt is unmasked in DSS_TPTC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
17	tptc_a0_write_access_error	R/W	0h	Status of Interrupt from DSS_TPTC_A0. Set only if Interrupt is unmasked in DSS_TPTC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
16	tpcc_a_write_access_error	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
15-9	RESERVED	R/W	0h	
8	tpcc_a_parity_err	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
7-4	RESERVED	R/W	0h	
3	tptc_a1_err	R/W	0h	Status of Interrupt from DSS_TPTC_A1. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
2	tptc_a0_err	R/W	0h	Status of Interrupt from DSS_TPTC_A0. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
1	tpcc_a_mpintr	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
0	tpcc_a_errintr	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interrupt is unmasked in DSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.

### 5.2.5.9 DSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW Register (Offset = 20h) [Reset = 00XXXXX0h]

DSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW is shown in [Table 5-417](#).

Return to the [Summary Table](#).

**Table 5-417. DSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	tptc_a1_read_access_error	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
25	tptc_a0_read_access_error	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
24	tpcc_a_read_access_error	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
23-19	RESERVED	R/W	0h	
18	tptc_a1_write_access_error	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in DSS_TPTC_A_INTAGG_MASK
17	tptc_a0_write_access_error	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in DSS_TPTC_A_INTAGG_MASK
16	tpcc_a_write_access_error	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
15-9	RESERVED	R/W	0h	
8	tpcc_a_parity_err	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
7-4	RESERVED	R/W	0h	
3	tptc_a1_err	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
2	tptc_a0_err	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
1	tpcc_a_mpint	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
0	tpcc_a_errint	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK

### 5.2.5.10 DSS\_TPCC\_A\_INTAGG\_MASK Register (Offset = 24h) [Reset = 0000XX00h]

DSS\_TPCC\_A\_INTAGG\_MASK is shown in [Table 5-418](#).

Return to the [Summary Table](#).

**Table 5-418. DSS\_TPCC\_A\_INTAGG\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	tptc_a1	R/W	0h	Mask Interrupt from DSS_TPTC_A1 to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	tptc_a0	R/W	0h	Mask Interrupt from DSS_TPTC_A0 to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	tpcc_a_int6	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	tpcc_a_int5	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	tpcc_a_int4	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	tpcc_a_int3	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	tpcc_a_int2	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	tpcc_a_int1	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	tpcc_a_int0	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	tpcc_a_intg	R/W	0h	Mask Interrupt from DSS_TPCC_A to aggregated Interrupt DSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

### 5.2.5.11 DSS\_TPCC\_A\_INTAGG\_STATUS Register (Offset = 28h) [Reset = 0000XX00h]

DSS\_TPCC\_A\_INTAGG\_STATUS is shown in [Table 5-419](#).

Return to the [Summary Table](#).

**Table 5-419. DSS\_TPCC\_A\_INTAGG\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	tptc_a1	R/W	0h	Status of Interrupt from DSS_TPTC_A1. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
16	tptc_a0	R/W	0h	Status of Interrupt from DSS_TPTC_A0. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
7	tpcc_a_int6	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
6	tpcc_a_int5	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
5	tpcc_a_int4	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
4	tpcc_a_int3	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
3	tpcc_a_int2	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
2	tpcc_a_int1	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
1	tpcc_a_int0	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
0	tpcc_a_intg	R/W	0h	Status of Interrupt from DSS_TPCC_A. Set only if Interupt is unmasked in DSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.

### 5.2.5.12 DSS\_TPCC\_A\_INTAGG\_STATUS\_RAW Register (Offset = 2Ch) [Reset = 0000XX00h]

DSS\_TPCC\_A\_INTAGG\_STATUS\_RAW is shown in [Table 5-420](#).

Return to the [Summary Table](#).

**Table 5-420. DSS\_TPCC\_A\_INTAGG\_STATUS\_RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	tptc_a1	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
16	tptc_a0	R/W	0h	Raw Status of Interrupt from DSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
7	tpcc_a_int6	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
6	tpcc_a_int5	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
5	tpcc_a_int4	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
4	tpcc_a_int3	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
3	tpcc_a_int2	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
2	tpcc_a_int1	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
1	tpcc_a_int0	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK
0	tpcc_a_intg	R/W	0h	Raw Status of Interrupt from DSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in DSS_TPCC_A_INTAGG_MASK

### 5.2.5.13 DSS\_TPCC\_MEMINIT\_START Register (Offset = 60h) [Reset = 00000000h]

DSS\_TPCC\_MEMINIT\_START is shown in [Table 5-421](#).

Return to the [Summary Table](#).

**Table 5-421. DSS\_TPCC\_MEMINIT\_START Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	tpcc_c_meminit_start	R/W	0h	RESERVED: Dont Use
1	tpcc_b_meminit_start	R/W	0h	RESERVED: Dont Use
0	tpcc_a_meminit_start	R/W	0h	Write pulse bit field: Start Memory initialization of TPCC A Param memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed. Before starting new initialization sequence ensure that there is no initialization sequence is in progress (ie TPCC_A_MEMINIT_STATUS should be 0x0) and clear any previous completion status (ie write 0x1 to clear TPCC_A_MEMINIT_DONE)

#### 5.2.5.14 DSS\_TPCC\_MEMINIT\_STATUS Register (Offset = 64h) [Reset = 00000000h]

DSS\_TPCC\_MEMINIT\_STATUS is shown in [Table 5-422](#).

Return to the [Summary Table](#).

**Table 5-422. DSS\_TPCC\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	tpcc_c_meminit_status	R	0h	RESERVED: Dont Use
1	tpcc_b_meminit_status	R	0h	RESERVED: Dont Use
0	tpcc_a_meminit_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is in progress.



### 5.2.5.15 DSS\_TPCC\_MEMINIT\_DONE Register (Offset = 68h) [Reset = 00000000h]

DSS\_TPCC\_MEMINIT\_DONE is shown in [Table 5-423](#).

Return to the [Summary Table](#).

**Table 5-423. DSS\_TPCC\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	tpcc_c_meminit_done	R/W	0h	RESERVED: Dont Use
1	tpcc_b_meminit_done	R/W	0h	RESERVED: Dont Use
0	tpcc_a_meminit_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is complte. Write 0x1 to clear status. Refer TPCC Memory initialization sequeunce in EDMA section for more details

### 5.2.5.16 DSS\_L3RAM\_MEMINIT\_START Register (Offset = 98h) [Reset = 00000000h]

DSS\_L3RAM\_MEMINIT\_START is shown in [Table 5-424](#).

Return to the [Summary Table](#).

**Table 5-424. DSS\_L3RAM\_MEMINIT\_START Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	l3ram3_meminit_start	R/W	0h	Start Memory initialization of TPCC A Param memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed. Before starting new initialization sequence ensure that there is no initialization sequence is in progress (ie L3RAM_MEMINIT_STATUS should be 0x0) and clear any previous completion status (ie write 0x1 to clear L3RAM_MEMINIT_DONE)
2	l3ram2_meminit_start	R/W	0h	Write pulse bit field: Start Memory initialization of TPCC A Param memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed. Before starting new initialization sequence ensure that there is no initialization sequence is in progress (ie L3RAM_MEMINIT_STATUS should be 0x0) and clear any previous completion status (ie write 0x1 to clear L3RAM_MEMINIT_DONE)
1	l3ram1_meminit_start	R/W	0h	Write pulse bit field: Start Memory initialization of TPCC A Param memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed. Before starting new initialization sequence ensure that there is no initialization sequence is in progress (ie L3RAM_MEMINIT_STATUS should be 0x0) and clear any previous completion status (ie write 0x1 to clear L3RAM_MEMINIT_DONE)
0	l3ram0_meminit_start	R/W	0h	Write pulse bit field: Start Memory initialization of TPCC A Param memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed. Before starting new initialization sequence ensure that there is no initialization sequence is in progress (ie L3RAM_MEMINIT_STATUS should be 0x0) and clear any previous completion status (ie write 0x1 to clear L3RAM_MEMINIT_DONE)

### 5.2.5.17 DSS\_L3RAM\_MEMINIT\_STATUS Register (Offset = 9Ch) [Reset = 00000000h]

DSS\_L3RAM\_MEMINIT\_STATUS is shown in [Table 5-425](#).

Return to the [Summary Table](#).

**Table 5-425. DSS\_L3RAM\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	l3ram3_meminit_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is in progress.
2	l3ram2_meminit_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is in progress.
1	l3ram1_meminit_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is in progress.
0	l3ram0_meminit_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is in progress.

### 5.2.5.18 DSS\_L3RAM\_MEMINIT\_DONE Register (Offset = A0h) [Reset = 00000000h]

DSS\_L3RAM\_MEMINIT\_DONE is shown in [Table 5-426](#).

Return to the [Summary Table](#).

**Table 5-426. DSS\_L3RAM\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	l3ram3_meminit_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is complete. Write 0x1 to clear status. Refer TPCC Memory initialization sequence in EDMA section for more details
2	l3ram2_meminit_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is complete. Write 0x1 to clear status. Refer TPCC Memory initialization sequence in EDMA section for more details
1	l3ram1_meminit_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is complete. Write 0x1 to clear status. Refer TPCC Memory initialization sequence in EDMA section for more details
0	l3ram0_meminit_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of TPCC A Param memory is complete. Write 0x1 to clear status. Refer TPCC Memory initialization sequence in EDMA section for more details

### 5.2.5.19 DSS\_TPCC\_A\_PARITY\_CTRL Register (Offset = BCh) [Reset = 00000000h]

DSS\_TPCC\_A\_PARITY\_CTRL is shown in [Table 5-427](#).

Return to the [Summary Table](#).

**Table 5-427. DSS\_TPCC\_A\_PARITY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	parity_err_clr	R/W	0h	Write pulse bit field: Write 0x1 to clear the Parit Error status for TPCC
1	parity_testen	R/W	0h	Enable Parity Test for TPCC. Write 0x1 : Parity Test is enabled on PARAM memory
0	parity_en	R/W	0h	Enable Parity for TPCC. Write 0x1 : Parity is enabled on PARAM memory

### 5.2.5.20 DSS\_TPCC\_A\_PARITY\_STATUS Register (Offset = C8h) [Reset = 00000000h]

DSS\_TPCC\_A\_PARITY\_STATUS is shown in [Table 5-428](#).

Return to the [Summary Table](#).

**Table 5-428. DSS\_TPCC\_A\_PARITY\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	parity_addr	R	0h	TPCC Error Address at which Parity Error occurred

### 5.2.5.21 TPTC\_DBS\_CONFIG Register (Offset = D4h) [Reset = 000AAAAAh]

TPTC\_DBS\_CONFIG is shown in [Table 5-429](#).

Return to the [Summary Table](#).

**Table 5-429. TPTC\_DBS\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-18	tptc_c5	R/W	2h	RESERVED: Dont Use
17-16	tptc_c4	R/W	2h	RESERVED: Dont Use
15-14	tptc_c3	R/W	2h	RESERVED: Dont Use
13-12	tptc_c2	R/W	2h	RESERVED: Dont Use
11-10	tptc_c1	R/W	2h	RESERVED: Dont Use
9-8	tptc_c0	R/W	2h	RESERVED: Dont Use0
7-6	tptc_b1	R/W	2h	RESERVED: Dont Use
5-4	tptc_b0	R/W	2h	RESERVED: Dont Use
3-2	tptc_a1	R/W	2h	Max Burst size tieoff value for TPTC A1
1-0	tptc_a0	R/W	2h	Max Burst size tieoff value for TPTC A0

### 5.2.5.22 DSS\_DSP\_BOOTCFG Register (Offset = D8h) [Reset = 00X02000h]

DSS\_DSP\_BOOTCFG is shown in [Table 5-430](#).

Return to the [Summary Table](#).

**Table 5-430. DSS\_DSP\_BOOTCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	L1P_CACHE_MODE	R/W	0h	DSP Boot Configuration : L1P Cache Mode
24	L1D_CACHE_MODE	R/W	0h	DSP Boot Configuration : L1D Cache Mode
23-22	RESERVED	R/W	0h	
21-0	ISTP_RST_VAL	R/W	2000h	DSP Boot Configuration : Reset Vector



### 5.2.5.23 DSS\_DSP\_NMI\_GATE Register (Offset = DCh) [Reset = 00000000h]

DSS\_DSP\_NMI\_GATE is shown in [Table 5-431](#).

Return to the [Summary Table](#).

**Table 5-431. DSS\_DSP\_NMI\_GATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	gate	R/W	0h	Write 3b111 to gate the Non Maskable Interrupt to the DSP. This is not expected to be used

#### 5.2.5.24 DSS\_PBIIST\_KEY\_RESET Register (Offset = E0h) [Reset = 00000000h]

DSS\_PBIIST\_KEY\_RESET is shown in [Table 5-432](#).

Return to the [Summary Table](#).

**Table 5-432. DSS\_PBIIST\_KEY\_RESET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	dss_l2_pbist_st_key	R/W	0h	DSS L2 PBIST Selftest Key.
7-4	dss_pbist_st_reset	R/W	0h	DSS PBIST controller will be brought out of reset when value is 0xA
3-0	dss_pbist_st_key	R/W	0h	DSS PBIST Selftest Key. Valid value is 0x5

### 5.2.5.25 DSS\_PBIIST\_REG0 Register (Offset = E4h) [Reset = 00000000h]

DSS\_PBIIST\_REG0 is shown in [Table 5-433](#).

Return to the [Summary Table](#).

**Table 5-433. DSS\_PBIIST\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	dss_pbist_reg0	R/W	0h	DSP PBIST registers

### 5.2.5.26 DSS\_PBIIST\_REG1 Register (Offset = E8h) [Reset = 00000000h]

DSS\_PBIIST\_REG1 is shown in [Table 5-434](#).

Return to the [Summary Table](#).

**Table 5-434. DSS\_PBIIST\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	dss_pbist_reg1	R/W	0h	DSP PBIST registers

### 5.2.5.27 DSS\_TPTC\_BOUNDARY\_CFG0 Register (Offset = ECh) [Reset = 12X2X2X2h]

DSS\_TPTC\_BOUNDARY\_CFG0 is shown in [Table 5-435](#).

Return to the [Summary Table](#).

**Table 5-435. DSS\_TPTC\_BOUNDARY\_CFG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	tptc_b1_size	R/W	12h	RESERVED: Dont Use
23-22	RESERVED	R/W	0h	
21-16	tptc_b0_size	R/W	12h	RESERVED: Dont Use
15-14	RESERVED	R/W	0h	
13-8	tptc_a1_size	R/W	12h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC Example: writing 6d19 decides boundary to be $2^{19}$ i.e. 512 KB
7-6	RESERVED	R/W	0h	
5-0	tptc_a0_size	R/W	12h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC Example: writing 6d19 decides boundary to be $2^{19}$ i.e. 512 KB

### 5.2.5.28 DSS\_TPTC\_BOUNDARY\_CFG1 Register (Offset = F0h) [Reset = 12X2X2X2h]

DSS\_TPTC\_BOUNDARY\_CFG1 is shown in [Table 5-436](#).

Return to the [Summary Table](#).

**Table 5-436. DSS\_TPTC\_BOUNDARY\_CFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-24	tptc_c3_size	R/W	12h	RESERVED: Dont Use
23-22	RESERVED	R/W	0h	
21-16	tptc_c2_size	R/W	12h	RESERVED: Dont Use
15-14	RESERVED	R/W	0h	
13-8	tptc_c1_size	R/W	12h	RESERVED: Dont Use
7-6	RESERVED	R/W	0h	
5-0	tptc_c0_size	R/W	12h	RESERVED: Dont Use

### 5.2.5.29 DSS\_TPTC\_BOUNDARY\_CFG2 Register (Offset = F4h) [Reset = 000012X2h]

DSS\_TPTC\_BOUNDARY\_CFG2 is shown in [Table 5-437](#).

Return to the [Summary Table](#).

**Table 5-437. DSS\_TPTC\_BOUNDARY\_CFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R/W	0h	
13-8	tptc_c5_size	R/W	12h	RESERVED: Dont Use
7-6	RESERVED	R/W	0h	
5-0	tptc_c4_size	R/W	12h	RESERVED: Dont Use

### 5.2.5.30 DSS\_TPTC\_XID\_REORDER\_CFG0 Register (Offset = F8h) [Reset = 00XXXXXXh]

DSS\_TPTC\_XID\_REORDER\_CFG0 is shown in [Table 5-438](#).

Return to the [Summary Table](#).

**Table 5-438. DSS\_TPTC\_XID\_REORDER\_CFG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	tptc_b1_disable	R/W	0h	RESERVED: Dont Use
23-17	RESERVED	R/W	0h	
16	tptc_b0_disable	R/W	0h	RESERVED: Dont Use
15-9	RESERVED	R/W	0h	
8	tptc_a1_disable	R/W	0h	Writing 1b1 will disable the CID-RID-SID reordering feature for the TPTC instance
7-1	RESERVED	R/W	0h	
0	tptc_a0_disable	R/W	0h	Writing 1b1 will disable the CID-RID-SID reordering feature for the TPTC instance



### 5.2.5.31 DSS\_TPTC\_XID\_REORDER\_CFG1 Register (Offset = FCh) [Reset = 00XXXXXXh]

DSS\_TPTC\_XID\_REORDER\_CFG1 is shown in [Table 5-439](#).

Return to the [Summary Table](#).

**Table 5-439. DSS\_TPTC\_XID\_REORDER\_CFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	tptc_c3_disable	R/W	0h	RESERVED: Dont Use
23-17	RESERVED	R/W	0h	
16	tptc_c2_disable	R/W	0h	RESERVED: Dont Use
15-9	RESERVED	R/W	0h	
8	tptc_c1_disable	R/W	0h	RESERVED: Dont Use
7-1	RESERVED	R/W	0h	
0	tptc_c0_disable	R/W	0h	RESERVED: Dont Use

### 5.2.5.32 DSS\_TPTC\_XID\_REORDER\_CFG2 Register (Offset = 100h) [Reset = 000000XXh]

DSS\_TPTC\_XID\_REORDER\_CFG2 is shown in [Table 5-440](#).

Return to the [Summary Table](#).

**Table 5-440. DSS\_TPTC\_XID\_REORDER\_CFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	tptc_c5_disable	R/W	0h	RESERVED: Dont Use
7-1	RESERVED	R/W	0h	
0	tptc_c4_disable	R/W	0h	RESERVED: Dont Use

### 5.2.5.33 ESM\_GATING0 Register (Offset = 108h) [Reset = FFFFFFFFh]

ESM\_GATING0 is shown in [Table 5-441](#).

Return to the [Summary Table](#).

**Table 5-441. ESM\_GATING0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	Bit3:0 : writing 000 will ungate the DSS ESM Group2 Error 0 Bit7:4 : writing 000 will ungate the DSS ESM Group2 Error 1 Bit31:28 : writing 000 will ungate the DSS ESM Group2 Error 7

### 5.2.5.34 ESM\_GATING1 Register (Offset = 10Ch) [Reset = FFFFFFFFh]

ESM\_GATING1 is shown in [Table 5-442](#).

Return to the [Summary Table](#).

**Table 5-442. ESM\_GATING1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	Bit3:0 : writing 000 will ungate the DSS ESM Group2 Error 8 Bit7:4 : writing 000 will ungate the DSS ESM Group2 Error 9 Bit31:28 : writing 000 will ungate the DSS ESM Group2 Error 15

### 5.2.5.35 ESM\_GATING2 Register (Offset = 110h) [Reset = FFFFFFFFh]

ESM\_GATING2 is shown in [Table 5-443](#).

Return to the [Summary Table](#).

**Table 5-443. ESM\_GATING2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	Bit3:0 : writing 000 will ungate the DSS ESM Group2 Error 16 Bit7:4 : writing 000 will ungate the DSS ESM Group2 Error 17 Bit31:28 : writing 000 will ungate the DSS ESM Group2 Error 23

### 5.2.5.36 ESM\_GATING3 Register (Offset = 114h) [Reset = FFFFFFFFh]

ESM\_GATING3 is shown in [Table 5-444](#).

Return to the [Summary Table](#).

**Table 5-444. ESM\_GATING3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	Bit3:0 : writing 000 will ungate the DSS ESM Group2 Error 24 Bit7:4 : writing 000 will ungate the DSS ESM Group2 Error 25 Bit31:28 : writing 000 will ungate the DSS ESM Group2 Error 31

### 5.2.5.37 DSS\_PERIPH\_ERRAGG\_MASK0 Register (Offset = 560h) [Reset = 00000000h]

DSS\_PERIPH\_ERRAGG\_MASK0 is shown in [Table 5-445](#).

Return to the [Summary Table](#).

**Table 5-445. DSS\_PERIPH\_ERRAGG\_MASK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	rcss_ctrl_wr	R/W	0h	Mask the Write error from RCSS_CTRL space from generating an error DSS_PERIPH_ERRAGG to the Processor
10	rcss_ctrl_rd	R/W	0h	Mask the Read error from RCSS_CTRL space from generating an error DSS_PERIPH_ERRAGG to the Processor
9	rcss_rcm_wr	R/W	0h	Mask the Write error from RCSS_RCM space from generating an error DSS_PERIPH_ERRAGG to the Processor
8	rcss_rcm_rd	R/W	0h	Mask the Read error from RCSS_RCM space from generating an error DSS_PERIPH_ERRAGG to the Processor
7	dss_hwa_cfg_wr	R/W	0h	Mask the Write error from DSS_HWA_CFG space from generating an error DSS_PERIPH_ERRAGG to the Processor
6	dss_hwa_cfg_rd	R/W	0h	Mask the Read error from DSS_HWA_CFG space from generating an error DSS_PERIPH_ERRAGG to the Processor
5	dss_cm4_ctrl_wr	R/W	0h	Mask the Write error from DSS_CM4_CTRL space from generating an error DSS_PERIPH_ERRAGG to the Processor
4	dss_cm4_ctrl_rd	R/W	0h	Mask the Read error from DSS_CM4_CTRL space from generating an error DSS_PERIPH_ERRAGG to the Processor
3	dss_ctrl_wr	R/W	0h	Mask the Write error from DSS_CTRL space from generating an error DSS_PERIPH_ERRAGG to the Processor
2	dss_ctrl_rd	R/W	0h	Mask the Read error from DSS_CTRL space from generating an error DSS_PERIPH_ERRAGG to the Processor
1	dss_rcm_wr	R/W	0h	Mask the Write error from DSS_RCM space from generating an error DSS_PERIPH_ERRAGG to the Processor
0	dss_rcm_rd	R/W	0h	Mask the Read error from DSS_RCM space from generating an error DSS_PERIPH_ERRAGG to the Processor

### 5.2.5.38 DSS\_PERIPH\_ERRAGG\_STATUS0 Register (Offset = 564h) [Reset = 00000000h]

DSS\_PERIPH\_ERRAGG\_STATUS0 is shown in [Table 5-446](#).

Return to the [Summary Table](#).

**Table 5-446. DSS\_PERIPH\_ERRAGG\_STATUS0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	rcss_ctrl_wr	R/W	0h	Status of the Write error from RCSS_CTRL space. Read 1 : Read error occurred on access to the RCSS_CTRL space
10	rcss_ctrl_rd	R/W	0h	Status of the Read error from RCSS_CTRL space. Read 1 : Read error occurred on access to the RCSS_CTRL space
9	rcss_rcm_wr	R/W	0h	Status of the Write error from RCSS_RCM space. Read 1 : Read error occurred on access to the RCSS_RCM space
8	rcss_rcm_rd	R/W	0h	Status of the Read error from RCSS_RCM space. Read 1 : Read error occurred on access to the RCSS_RCM space
7	dss_hwa_cfg_wr	R/W	0h	Status of the Write error from DSS_HWA_CFG space. Read 1 : Read error occurred on access to the DSS_HWA_CFG space
6	dss_hwa_cfg_rd	R/W	0h	Status of the Read error from DSS_HWA_CFG space. Read 1 : Read error occurred on access to the DSS_HWA_CFG space
5	dss_cm4_ctrl_wr	R/W	0h	Status of the Write error from DSS_CM4_CTRL space. Read 1 : Read error occurred on access to the DSS_CM4_CTRL space
4	dss_cm4_ctrl_rd	R/W	0h	Status of the Read error from DSS_CM4_CTRL space. Read 1 : Read error occurred on access to the DSS_CM4_CTRL space
3	dss_ctrl_wr	R/W	0h	Status of the Write error from DSS_CTRL space. Read 1 : Read error occurred on access to the DSS_CTRL space
2	dss_ctrl_rd	R/W	0h	Status of the Read error from DSS_CTRL space. Read 1 : Read error occurred on access to the DSS_CTRL space
1	dss_rcm_wr	R/W	0h	Status of the Write error from DSS_RCM space. Read 1 : Read error occurred on access to the DSS_RCM space
0	dss_rcm_rd	R/W	0h	Status of the Read error from DSS_RCM space. Read 1 : Read error occurred on access to the DSS_RCM space



### 5.2.5.39 DSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 Register (Offset = 568h) [Reset = 00000000h]

DSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 is shown in [Table 5-447](#).

Return to the [Summary Table](#).

**Table 5-447. DSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11	rcss_ctrl_wr	R/W	0h	Raw Status of the Write error from RCSS_CTRL space irrespective of it being masked. Read 1 : Read error occurred on access to the RCSS_CTRL space
10	rcss_ctrl_rd	R/W	0h	Raw Status of the Read error from RCSS_CTRL space irrespective of it being masked. Read 1 : Read error occurred on access to the RCSS_CTRL space
9	rcss_rcm_wr	R/W	0h	Raw Status of the Write error from RCSS_RCM space irrespective of it being masked. Read 1 : Read error occurred on access to the RCSS_RCM space
8	rcss_rcm_rd	R/W	0h	Raw Status of the Read error from RCSS_RCM space irrespective of it being masked. Read 1 : Read error occurred on access to the RCSS_RCM space
7	dss_hwa_cfg_wr	R/W	0h	Raw Status of the Write error from DSS_HWA_CFG space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_HWA_CFG space
6	dss_hwa_cfg_rd	R/W	0h	Raw Status of the Read error from DSS_HWA_CFG space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_HWA_CFG space
5	dss_cm4_ctrl_wr	R/W	0h	Raw Status of the Write error from DSS_CM4_CTRL space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_CM4_CTRL space
4	dss_cm4_ctrl_rd	R/W	0h	Raw Status of the Read error from DSS_CM4_CTRL space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_CM4_CTRL space
3	dss_ctrl_wr	R/W	0h	Raw Status of the Write error from DSS_CTRL space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_CTRL space
2	dss_ctrl_rd	R/W	0h	Raw Status of the Read error from DSS_CTRL space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_CTRL space
1	dss_rcm_wr	R/W	0h	Raw Status of the Write error from DSS_RCM space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_RCM space
0	dss_rcm_rd	R/W	0h	Raw Status of the Read error from DSS_RCM space irrespective of it being masked. Read 1 : Read error occurred on access to the DSS_RCM space

### 5.2.5.40 DBG\_ACK\_CPU\_CTRL Register (Offset = 584h) [Reset = 00000000h]

DBG\_ACK\_CPU\_CTRL is shown in [Table 5-448](#).

Return to the [Summary Table](#).

**Table 5-448. DBG\_ACK\_CPU\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	sel	R/W	0h	Select the Processor Suspend that is used to Suspend the DSS Peripherals 0: DSP 1:MSS CR5

### 5.2.5.41 DBG\_ACK\_CTL0 Register (Offset = 588h) [Reset = 000XXXXXh]

DBG\_ACK\_CTL0 is shown in [Table 5-449](#).

Return to the [Summary Table](#).

**Table 5-449. DBG\_ACK\_CTL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-20	DSS_WDT	R/W	0h	RESERVED: Dont Use
19	RESERVED	R/W	0h	
18-16	DSS_SCIA	R/W	0h	RESERVED: Dont Use
15	RESERVED	R/W	0h	
14-12	DSS_RTIB	R/W	0h	RESERVED: Dont Use
11	RESERVED	R/W	0h	
10-8	DSS_RTIA	R/W	0h	ERESERVED: Dont Use
7	RESERVED	R/W	0h	
6-4	DSS_DCCB	R/W	0h	RESERVED: Dont Use
3	RESERVED	R/W	0h	
2-0	DSS_DCCA	R/W	0h	RESERVED: Dont Use

### 5.2.5.42 DBG\_ACK\_CTL1 Register (Offset = 58Ch) [Reset = 0XXXXXXXh]

DBG\_ACK\_CTL1 is shown in [Table 5-450](#).

Return to the [Summary Table](#).

**Table 5-450. DBG\_ACK\_CTL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30-28	DSS_HWA	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
27	RESERVED	R/W	0h	
26-24	DSS_MCRC	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
23-0	RESERVED	R/W	0h	

#### 5.2.5.43 DSS\_DSP\_INT\_SEL Register (Offset = 590h) [Reset = 00000000h]

DSS\_DSP\_INT\_SEL is shown in [Table 5-451](#).

Return to the [Summary Table](#).

**Table 5-451. DSS\_DSP\_INT\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	RCSS_CSI2_ICSSM	R/W	0h	DSS DSP Interrupt selcet 0x0: CSI2 Interrupts are propagated to DSP 0x7 : ICSSM Interrupts are propagted to DSP

#### 5.2.5.44 DSS\_CBUFF\_TRIGGER\_SEL Register (Offset = 594h) [Reset = 00000028h]

DSS\_CBUFF\_TRIGGER\_SEL is shown in [Table 5-452](#).

Return to the [Summary Table](#).

**Table 5-452. DSS\_CBUFF\_TRIGGER\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	
6-0	sel	R/W	28h	DSS CBUFF HW Trigger select from DSS DSP Interrupt Map. Reset value selects RSS_ADC_CAPTURE_COMPLETE as cbuff trigger

### 5.2.5.45 CSI2\_TX\_PARSTATCFG Register (Offset = CE8h) [Reset = 000000X0h]

CSI2\_TX\_PARSTATCFG is shown in [Table 5-453](#).

Return to the [Summary Table](#).

**Table 5-453. CSI2\_TX\_PARSTATCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	sten	R/W	0h	Enable bit for the self test of the Parity logic in CSI2
9	enable	R/W	0h	Enable bit for the Parity computation in CSI2
8	clr	R/W	0h	Clear bit for the Parity error from CSI2
7	RESERVED	R/W	0h	
6-0	stat	R	0h	Parity address from CSI2

### 5.2.5.46 CSI2\_TX\_CFG1 Register (Offset = CECh) [Reset = 00FBFE7Fh]

CSI2\_TX\_CFG1 is shown in [Table 5-454](#).

Return to the [Summary Table](#).

**Table 5-454. CSI2\_TX\_CFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CIOUSELDOEXTERNAL	R	0h	Connected to dsi_protocol port name csi_ciouseldoexternal
30	CIOLDOPWRGOOD	R/W	0h	Connected to dsi_protocol port name csi_cioldopwrgood
29	DISPCUPDATESYNC	R/W	0h	Connected to dsi_protocol port name csi_dispcupdatesync
28	TE1LINE	R/W	0h	Connected to dsi_protocol port name csi_te1line
27	TE0LINE	R/W	0h	Connected to dsi_protocol port name csi_te0line
26	BYPASSEN	R/W	0h	Connected to dsi_protocol port name csi_bypassen
25	HHV	R/W	0h	Connected to dsi_protocol port name csi_hhv
24	OFFMODE	R/W	0h	Connected to dsi_protocol port name csi_offmode
23	PIPD4	R/W	1h	Connected to dsi_protocol port name csi_pipd4
22	PIPD3	R/W	1h	Connected to dsi_protocol port name csi_pipd3
21	PIPD2	R/W	1h	Connected to dsi_protocol port name csi_pipd2
20	PIPD1	R/W	1h	Connected to dsi_protocol port name csi_pipd1
19	PIPD0	R/W	1h	Connected to dsi_protocol port name csi_pipd0
18	SWAKEUP	R	0h	Connected to dsi_protocol port name csi_po_wakeup
17	SIDLEACK	R	1h	Connected to dsi_protocol port name csi_po_sidleack
16-12	LANEENABLE	R/W	1Fh	Connected to dsi_protocol port name csi_laneenable
11-10	NUMOFLANES	R/W	3h	Connected to dsi_protocol port name csi_numoflanes
9	MIDLEREQ	R/W	1h	Connected to dsi_protocol port name csi_pi_midlereq
8	VPCLKEN	R/W	0h	Connected to dsi_protocol port name mem_vp_clk_en
7	VPPCLKEN	R/W	0h	Connected to dsi_protocol port name mem_vp_pclk_en
6	SCPBUSY	R/W	1h	Connected to dsi_protocol port name csi_pi_scpbusy
5	TXBYTECLKHS	R/W	1h	Connected to dsi_protocol port name csi_pi_top_clkdiv_txbyteclkhs
4-3	TOPCLKDIVDSI	R/W	3h	Connected to dsi_protocol port name csi_pi_top_clkdiv_dsi
2-1	TOPCLKDIVL3SCP	R/W	3h	Connected to dsi_protocol port name csi_pi_top_clkdiv_l3scp
0	OCPCLKEN	R/W	1h	Connected to dsi_protocol port name csi_pi_ocp_clken



### 5.2.5.47 CSI2\_TX\_CFG2 Register (Offset = CF0h) [Reset = 0000XXAh]

CSI2\_TX\_CFG2 is shown in [Table 5-455](#).

Return to the [Summary Table](#).

**Table 5-455. CSI2\_TX\_CFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	DSIADDRREMAPEN	R/W	0h	Enable to Mux the re-mapped addresses for bypass access path to CSI. 000 -->Disable, 111-->Enable, Others are reserved.
28-26	CSIBURSTEN	R/W	0h	Enable to Mux the Burst related signals onto the MSB address bits to the VBUS2OCP module. 000 -->Disable, 111-->Enable, Others are reserved.
25-23	CSIB2BWREADYEN	R/W	0h	Enable for the burst mode support of VBUSP 2 OCP module. 000 -->Disable, 111-->Enable, Others are reserved.
22-20	PORCLKSELOVRVAL	R/W	0h	Override value for CLKLANEADDR port of DSI PHY.
19	PORCLKSELOVRCTL	R/W	0h	Override control for CLKLANEADDR port of DSI PHY.
18	CLKINENOVRRVAL	R/W	0h	Override value for CLKINEN port of DSI PHY. Invert of this value is connected to DSI PHY
17	CLKINENOVRRCTL	R/W	0h	Override control for CLKINEN port of DSI PHY
16	BYPASSACKZ	R	0h	Connected to DSI PHY csi_bypassackz
15	PLLPWRCMDONALL	R	0h	Connected to dsi_protocol port name csi_pllpwrcmdonall
14	PLLPWRCMDONDIV	R	0h	Connected to dsi_protocol port name csi_pllpwrcmdondiv
13	PLLPWRCMDONHSCLK	R	0h	Connected to dsi_protocol port name csi_pllpwrcmdonhsclock
12	PLLPWRCMDOFF	R	0h	Connected to dsi_protocol port name csi_pllpwrcmdoff
11-7	RESERVED	R/W	0h	
6-4	CLKIN4DDRGGOODBAR	R/W	0h	Connected to DSI PHY csi_clkin4ddrgoodbar
3	PLLPWRRACK	R/W	1h	Connected to dsi_protocol port name csi_pllpwrrack
2	DSIRECAL	R/W	0h	Connected to dsi_protocol port name csi_dsirecal
1	DSILOCK	R/W	1h	Connected to dsi_protocol port name csi_dsilock
0	SCPINPLLCRTL	R/W	0h	Connected to dsi_protocol port name csi_scpinpllcrtl

### 5.2.5.48 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-456](#).

Return to the [Summary Table](#).

**Table 5-456. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.5.49 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-457](#).

Return to the [Summary Table](#).

**Table 5-457. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.5.50 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-458](#).

Return to the [Summary Table](#).

**Table 5-458. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D

### 5.2.5.51 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-459](#).

Return to the [Summary Table](#).

**Table 5-459. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.5.52 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-460](#).

Return to the [Summary Table](#).

**Table 5-460. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.5.53 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-461](#).

Return to the [Summary Table](#).

**Table 5-461. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

### 5.2.5.54 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-462](#).

Return to the [Summary Table](#).

**Table 5-462. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D



### 5.2.5.55 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-463](#).

Return to the [Summary Table](#).

**Table 5-463. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.5.56 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-464](#).

Return to the [Summary Table](#).

**Table 5-464. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.5.57 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-465](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-465. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

### 5.2.5.58 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-466](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-466. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.5.59 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-467](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-467. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.5.60 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-468](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-468. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.5.61 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-469](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-469. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.5.62 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-470](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-470. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.



### 5.2.5.63 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-471](#).

Return to the [Summary Table](#).

EOI register

**Table 5-471. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.5.64 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-472](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-472. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.5.65 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-473](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-473. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault

### 5.2.5.66 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-474](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-474. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.5.67 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-475](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-475. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

## 5.2.6 RSS\_CTRL Registers

Table 5-476 lists the memory-mapped registers for the RSS\_CTRL registers. All register offset addresses not listed in Table 5-476 should be considered as reserved locations and the register contents should not be modified.

**Table 5-476. RSS\_CTRL Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.6.1</a>
8h	RSS_TPCC_A_ERRAGG_MASK		<a href="#">Section 5.2.6.2</a>
Ch	RSS_TPCC_A_ERRAGG_STATUS		<a href="#">Section 5.2.6.3</a>
10h	RSS_TPCC_A_ERRAGG_STATUS_RAW		<a href="#">Section 5.2.6.4</a>
14h	RSS_TPCC_A_INTAGG_MASK		<a href="#">Section 5.2.6.5</a>
18h	RSS_TPCC_A_INTAGG_STATUS		<a href="#">Section 5.2.6.6</a>
1Ch	RSS_TPCC_A_INTAGG_STATUS_RAW		<a href="#">Section 5.2.6.7</a>
20h	RSS_TPCC_MEMINIT_START		<a href="#">Section 5.2.6.8</a>
24h	RSS_TPCC_MEMINIT_DONE		<a href="#">Section 5.2.6.9</a>
28h	RSS_TPCC_MEMINIT_STATUS		<a href="#">Section 5.2.6.10</a>
2Ch	TPTC_DBS_CFG		<a href="#">Section 5.2.6.11</a>
30h	RSS_TPCC_A_PARITY_CTRL		<a href="#">Section 5.2.6.12</a>
34h	RSS_TPCC_A_PARITY_STATUS		<a href="#">Section 5.2.6.13</a>
274h	RSS_TPTC_BOUNDARY_CFG		<a href="#">Section 5.2.6.14</a>
278h	RSS_TPTC_XID_REORDER_CFG		<a href="#">Section 5.2.6.15</a>
27Ch	DBG_ACK_CPU_CTRL		<a href="#">Section 5.2.6.16</a>
280h	RSS_ADCBUF_PING_MEMINIT		<a href="#">Section 5.2.6.17</a>
284h	RSS_ADCBUF_PING_MEMINIT_DONE		<a href="#">Section 5.2.6.18</a>
288h	RSS_ADCBUF_PING_MEMINIT_STATUS		<a href="#">Section 5.2.6.19</a>
28Ch	RSS_ADCBUF_PONG_MEMINIT		<a href="#">Section 5.2.6.20</a>
290h	RSS_ADCBUF_PONG_MEMINIT_DONE		<a href="#">Section 5.2.6.21</a>
294h	RSS_ADCBUF_PONG_MEMINIT_STATU S		<a href="#">Section 5.2.6.22</a>
2C8h	SOC_TO_BSS_SW_INT		<a href="#">Section 5.2.6.23</a>
2CCh	RSS_DBG_ACK_CTL0		<a href="#">Section 5.2.6.24</a>
2D0h	DMMSWINT1		<a href="#">Section 5.2.6.25</a>
2D4h	RSS_SHARED_MEM_MEMINIT		<a href="#">Section 5.2.6.26</a>
2D8h	RSS_SHARED_MEM_MEMINIT_DONE		<a href="#">Section 5.2.6.27</a>
2DCh	RSS_SHARED_MEM_MEMINIT_STATUS		<a href="#">Section 5.2.6.28</a>
400h	BSS_CONTROL		<a href="#">Section 5.2.6.29</a>
404h	BSS_TCM_MEMINIT		<a href="#">Section 5.2.6.30</a>
408h	BSS_TCM_MEMINIT_DONE		<a href="#">Section 5.2.6.31</a>
40Ch	BSS_TCM_MEMINIT_STATUS		<a href="#">Section 5.2.6.32</a>
410h	BSS_VIM_MEMINIT		<a href="#">Section 5.2.6.33</a>
414h	BSS_VIM_MEMINIT_DONE		<a href="#">Section 5.2.6.34</a>
418h	BSS_VIM_MEMINIT_STATUS		<a href="#">Section 5.2.6.35</a>
41Ch	BSS_DFE_MEMINIT		<a href="#">Section 5.2.6.36</a>
420h	BSS_DFE_MEMINIT_DONE		<a href="#">Section 5.2.6.37</a>
424h	BSS_DFE_MEMINIT_STATUS		<a href="#">Section 5.2.6.38</a>
428h	BSS_RAMPGEN_MEMINIT		<a href="#">Section 5.2.6.39</a>
42Ch	BSS_RAMPGEN_MEMINIT_DONE		<a href="#">Section 5.2.6.40</a>

**Table 5-476. RSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
430h	BSS_RAMPGEN_MEMINIT_STATUS		<a href="#">Section 5.2.6.41</a>
434h	BSS_DSS_L3_STICKY		<a href="#">Section 5.2.6.42</a>
438h	BSS_DSS_L3_ACCESS		<a href="#">Section 5.2.6.43</a>
800h	TESTPATTERNRX1ICFG		<a href="#">Section 5.2.6.44</a>
804h	TESTPATTERNRX2ICFG		<a href="#">Section 5.2.6.45</a>
808h	TESTPATTERNRX3ICFG		<a href="#">Section 5.2.6.46</a>
80Ch	TESTPATTERNRX4ICFG		<a href="#">Section 5.2.6.47</a>
810h	TESTPATTERNRX1QCFG		<a href="#">Section 5.2.6.48</a>
814h	TESTPATTERNRX2QCFG		<a href="#">Section 5.2.6.49</a>
818h	TESTPATTERNRX3QCFG		<a href="#">Section 5.2.6.50</a>
81Ch	TESTPATTERNRX4QCFG		<a href="#">Section 5.2.6.51</a>
820h	TESTPATTERNVLDCFG		<a href="#">Section 5.2.6.52</a>
824h	ADCBUFCFG1		<a href="#">Section 5.2.6.53</a>
828h	ADCBUFCFG1_EXTD		<a href="#">Section 5.2.6.54</a>
82Ch	ADCBUFCFG2		<a href="#">Section 5.2.6.55</a>
830h	ADCBUFCFG3		<a href="#">Section 5.2.6.56</a>
834h	ADCBUFCFG4		<a href="#">Section 5.2.6.57</a>
838h	ADCBUFINTGENDITHERDLY		<a href="#">Section 5.2.6.58</a>
83Ch	CBUFF_FRAME_START_SEL		<a href="#">Section 5.2.6.59</a>
C00h	CQCFG1		<a href="#">Section 5.2.6.60</a>
C04h	CQCFG2		<a href="#">Section 5.2.6.61</a>
C08h	CPREG0		<a href="#">Section 5.2.6.62</a>
C0Ch	CPREG1		<a href="#">Section 5.2.6.63</a>
C10h	CPREG2		<a href="#">Section 5.2.6.64</a>
C14h	CPREG3		<a href="#">Section 5.2.6.65</a>
C18h	CPREG4		<a href="#">Section 5.2.6.66</a>
C1Ch	CPREG5		<a href="#">Section 5.2.6.67</a>
C20h	CPREG6		<a href="#">Section 5.2.6.68</a>
C24h	CPREG7		<a href="#">Section 5.2.6.69</a>
C28h	CPREG8		<a href="#">Section 5.2.6.70</a>
C2Ch	CPREG9		<a href="#">Section 5.2.6.71</a>
C30h	CPREG10		<a href="#">Section 5.2.6.72</a>
C34h	CPREG11		<a href="#">Section 5.2.6.73</a>
C38h	CPREG12		<a href="#">Section 5.2.6.74</a>
C3Ch	CPREG13		<a href="#">Section 5.2.6.75</a>
C40h	CPREG14		<a href="#">Section 5.2.6.76</a>
C44h	CPREG15		<a href="#">Section 5.2.6.77</a>
C48h	CH0CPREG0		<a href="#">Section 5.2.6.78</a>
C4Ch	CH0CPREG1		<a href="#">Section 5.2.6.79</a>
C50h	CH0CPREG2		<a href="#">Section 5.2.6.80</a>
C54h	CH0CPREG3		<a href="#">Section 5.2.6.81</a>
C58h	CH0CPREG4		<a href="#">Section 5.2.6.82</a>
C5Ch	CH0CPREG5		<a href="#">Section 5.2.6.83</a>
C60h	CH0CPREG6		<a href="#">Section 5.2.6.84</a>
C64h	CH0CPREG7		<a href="#">Section 5.2.6.85</a>

**Table 5-476. RSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
C68h	CH0CPREG8		<a href="#">Section 5.2.6.86</a>
C6Ch	CH0CPREG9		<a href="#">Section 5.2.6.87</a>
C70h	CH0CPREG10		<a href="#">Section 5.2.6.88</a>
C74h	CH0CPREG11		<a href="#">Section 5.2.6.89</a>
C78h	CH0CPREG12		<a href="#">Section 5.2.6.90</a>
C7Ch	CH0CPREG13		<a href="#">Section 5.2.6.91</a>
C80h	CH0CPREG14		<a href="#">Section 5.2.6.92</a>
C84h	CH0CPREG15		<a href="#">Section 5.2.6.93</a>
C88h	CH1CPREG0		<a href="#">Section 5.2.6.94</a>
C8Ch	CH1CPREG1		<a href="#">Section 5.2.6.95</a>
C90h	CH1CPREG2		<a href="#">Section 5.2.6.96</a>
C94h	CH1CPREG3		<a href="#">Section 5.2.6.97</a>
C98h	CH1CPREG4		<a href="#">Section 5.2.6.98</a>
C9Ch	CH1CPREG5		<a href="#">Section 5.2.6.99</a>
CA0h	CH1CPREG6		<a href="#">Section 5.2.6.100</a>
CA4h	CH1CPREG7		<a href="#">Section 5.2.6.101</a>
CA8h	CH1CPREG8		<a href="#">Section 5.2.6.102</a>
CACH	CH1CPREG9		<a href="#">Section 5.2.6.103</a>
CB0h	CH1CPREG10		<a href="#">Section 5.2.6.104</a>
CB4h	CH1CPREG11		<a href="#">Section 5.2.6.105</a>
CB8h	CH1CPREG12		<a href="#">Section 5.2.6.106</a>
CBCh	CH1CPREG13		<a href="#">Section 5.2.6.107</a>
CC0h	CH1CPREG14		<a href="#">Section 5.2.6.108</a>
CC4h	CH1CPREG15		<a href="#">Section 5.2.6.109</a>
CC8h	CH2CPREG0		<a href="#">Section 5.2.6.110</a>
CCCh	CH2CPREG1		<a href="#">Section 5.2.6.111</a>
CD0h	CH2CPREG2		<a href="#">Section 5.2.6.112</a>
CD4h	CH2CPREG3		<a href="#">Section 5.2.6.113</a>
CD8h	CH2CPREG4		<a href="#">Section 5.2.6.114</a>
CDCh	CH2CPREG5		<a href="#">Section 5.2.6.115</a>
CE0h	CH2CPREG6		<a href="#">Section 5.2.6.116</a>
CE4h	CH2CPREG7		<a href="#">Section 5.2.6.117</a>
CE8h	CH2CPREG8		<a href="#">Section 5.2.6.118</a>
CECh	CH2CPREG9		<a href="#">Section 5.2.6.119</a>
CF0h	CH2CPREG10		<a href="#">Section 5.2.6.120</a>
CF4h	CH2CPREG11		<a href="#">Section 5.2.6.121</a>
CF8h	CH2CPREG12		<a href="#">Section 5.2.6.122</a>
CFCh	CH2CPREG13		<a href="#">Section 5.2.6.123</a>
D00h	CH2CPREG14		<a href="#">Section 5.2.6.124</a>
D04h	CH2CPREG15		<a href="#">Section 5.2.6.125</a>
D08h	CH3CPREG0		<a href="#">Section 5.2.6.126</a>
D0Ch	CH3CPREG1		<a href="#">Section 5.2.6.127</a>
D10h	CH3CPREG2		<a href="#">Section 5.2.6.128</a>
D14h	CH3CPREG3		<a href="#">Section 5.2.6.129</a>
D18h	CH3CPREG4		<a href="#">Section 5.2.6.130</a>



**Table 5-476. RSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
D1Ch	CH3CPREG5		<a href="#">Section 5.2.6.131</a>
D20h	CH3CPREG6		<a href="#">Section 5.2.6.132</a>
D24h	CH3CPREG7		<a href="#">Section 5.2.6.133</a>
D28h	CH3CPREG8		<a href="#">Section 5.2.6.134</a>
D2Ch	CH3CPREG9		<a href="#">Section 5.2.6.135</a>
D30h	CH3CPREG10		<a href="#">Section 5.2.6.136</a>
D34h	CH3CPREG11		<a href="#">Section 5.2.6.137</a>
D38h	CH3CPREG12		<a href="#">Section 5.2.6.138</a>
D3Ch	CH3CPREG13		<a href="#">Section 5.2.6.139</a>
D40h	CH3CPREG14		<a href="#">Section 5.2.6.140</a>
D44h	CH3CPREG15		<a href="#">Section 5.2.6.141</a>
D48h	CH4CPREG0		<a href="#">Section 5.2.6.142</a>
D4Ch	CH4CPREG1		<a href="#">Section 5.2.6.143</a>
D50h	CH4CPREG2		<a href="#">Section 5.2.6.144</a>
D54h	CH4CPREG3		<a href="#">Section 5.2.6.145</a>
D58h	CH4CPREG4		<a href="#">Section 5.2.6.146</a>
D5Ch	CH4CPREG5		<a href="#">Section 5.2.6.147</a>
D60h	CH4CPREG6		<a href="#">Section 5.2.6.148</a>
D64h	CH4CPREG7		<a href="#">Section 5.2.6.149</a>
D68h	CH4CPREG8		<a href="#">Section 5.2.6.150</a>
D6Ch	CH4CPREG9		<a href="#">Section 5.2.6.151</a>
D70h	CH4CPREG10		<a href="#">Section 5.2.6.152</a>
D74h	CH4CPREG11		<a href="#">Section 5.2.6.153</a>
D78h	CH4CPREG12		<a href="#">Section 5.2.6.154</a>
D7Ch	CH4CPREG13		<a href="#">Section 5.2.6.155</a>
D80h	CH4CPREG14		<a href="#">Section 5.2.6.156</a>
D84h	CH4CPREG15		<a href="#">Section 5.2.6.157</a>
D88h	CH5CPREG0		<a href="#">Section 5.2.6.158</a>
D8Ch	CH5CPREG1		<a href="#">Section 5.2.6.159</a>
D90h	CH5CPREG2		<a href="#">Section 5.2.6.160</a>
D94h	CH5CPREG3		<a href="#">Section 5.2.6.161</a>
D98h	CH5CPREG4		<a href="#">Section 5.2.6.162</a>
D9Ch	CH5CPREG5		<a href="#">Section 5.2.6.163</a>
DA0h	CH5CPREG6		<a href="#">Section 5.2.6.164</a>
DA4h	CH5CPREG7		<a href="#">Section 5.2.6.165</a>
DA8h	CH5CPREG8		<a href="#">Section 5.2.6.166</a>
DACh	CH5CPREG9		<a href="#">Section 5.2.6.167</a>
DB0h	CH5CPREG10		<a href="#">Section 5.2.6.168</a>
DB4h	CH5CPREG11		<a href="#">Section 5.2.6.169</a>
DB8h	CH5CPREG12		<a href="#">Section 5.2.6.170</a>
DBCh	CH5CPREG13		<a href="#">Section 5.2.6.171</a>
DC0h	CH5CPREG14		<a href="#">Section 5.2.6.172</a>
DC4h	CH5CPREG15		<a href="#">Section 5.2.6.173</a>
DC8h	CH6CPREG0		<a href="#">Section 5.2.6.174</a>
DCCh	CH6CPREG1		<a href="#">Section 5.2.6.175</a>

**Table 5-476. RSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
DD0h	CH6CPREG2		<a href="#">Section 5.2.6.176</a>
DD4h	CH6CPREG3		<a href="#">Section 5.2.6.177</a>
DD8h	CH6CPREG4		<a href="#">Section 5.2.6.178</a>
DDCh	CH6CPREG5		<a href="#">Section 5.2.6.179</a>
DE0h	CH6CPREG6		<a href="#">Section 5.2.6.180</a>
DE4h	CH6CPREG7		<a href="#">Section 5.2.6.181</a>
DE8h	CH6CPREG8		<a href="#">Section 5.2.6.182</a>
DECh	CH6CPREG9		<a href="#">Section 5.2.6.183</a>
DF0h	CH6CPREG10		<a href="#">Section 5.2.6.184</a>
DF4h	CH6CPREG11		<a href="#">Section 5.2.6.185</a>
DF8h	CH6CPREG12		<a href="#">Section 5.2.6.186</a>
DFCh	CH6CPREG13		<a href="#">Section 5.2.6.187</a>
E00h	CH6CPREG14		<a href="#">Section 5.2.6.188</a>
E04h	CH6CPREG15		<a href="#">Section 5.2.6.189</a>
E08h	CH7CPREG0		<a href="#">Section 5.2.6.190</a>
E0Ch	CH7CPREG1		<a href="#">Section 5.2.6.191</a>
E10h	CH7CPREG2		<a href="#">Section 5.2.6.192</a>
E14h	CH7CPREG3		<a href="#">Section 5.2.6.193</a>
E18h	CH7CPREG4		<a href="#">Section 5.2.6.194</a>
E1Ch	CH7CPREG5		<a href="#">Section 5.2.6.195</a>
E20h	CH7CPREG6		<a href="#">Section 5.2.6.196</a>
E24h	CH7CPREG7		<a href="#">Section 5.2.6.197</a>
E28h	CH7CPREG8		<a href="#">Section 5.2.6.198</a>
E2Ch	CH7CPREG9		<a href="#">Section 5.2.6.199</a>
E30h	CH7CPREG10		<a href="#">Section 5.2.6.200</a>
E34h	CH7CPREG11		<a href="#">Section 5.2.6.201</a>
E38h	CH7CPREG12		<a href="#">Section 5.2.6.202</a>
E3Ch	CH7CPREG13		<a href="#">Section 5.2.6.203</a>
E40h	CH7CPREG14		<a href="#">Section 5.2.6.204</a>
E44h	CH7CPREG15		<a href="#">Section 5.2.6.205</a>
E48h	CH01_HIL_CP_OVERRIDE		<a href="#">Section 5.2.6.206</a>
E4Ch	CH23_HIL_CP_OVERRIDE		<a href="#">Section 5.2.6.207</a>
E50h	CH45_HIL_CP_OVERRIDE		<a href="#">Section 5.2.6.208</a>
E54h	CH67_HIL_CP_OVERRIDE		<a href="#">Section 5.2.6.209</a>
E58h	CH_HIL_CP_OVERRIDE		<a href="#">Section 5.2.6.210</a>
E5Ch	RSS_BOOKKEEPING_CTRL		<a href="#">Section 5.2.6.211</a>
E60h	RSS_APP_GP		<a href="#">Section 5.2.6.212</a>
E64h	RSS_BOOKKEEPING_SEQ_NUM		<a href="#">Section 5.2.6.213</a>
E68h	RSS_BOOKKEEPING_FRM_CNT		<a href="#">Section 5.2.6.214</a>
E6Ch	RSS_BOOKKEEPING_CHRP_CNT		<a href="#">Section 5.2.6.215</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.6.216</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.6.217</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.6.218</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.6.219</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.6.220</a>

**Table 5-476. RSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.6.221</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.6.222</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.6.223</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.6.224</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.6.225</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.6.226</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.6.227</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.6.228</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.6.229</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.6.230</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.6.231</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.6.232</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.6.233</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.6.234</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.6.235</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.6.236</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-477](#) shows the codes that are used for access types in this section.

**Table 5-477. RSS\_CTRL Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.6.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-478](#).

Return to the [Summary Table](#).

PID register

**Table 5-478. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.6.2 RSS\_TPCC\_A\_ERRAGG\_MASK Register (Offset = 8h) [Reset = 00XXXXXXh]

RSS\_TPCC\_A\_ERRAGG\_MASK is shown in [Table 5-479](#).

Return to the [Summary Table](#).

**Table 5-479. RSS\_TPCC\_A\_ERRAGG\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	tptc_a0_read_access_error	R/W	0h	
24	tpcc_a_read_access_error	R/W	0h	
23-18	RESERVED	R/W	0h	
17	tptc_a0_write_access_error	R/W	0h	
16	tpcc_a_write_access_error	R/W	0h	
15-9	RESERVED	R/W	0h	
8	tpcc_a_parity_err	R/W	0h	
7-3	RESERVED	R/W	0h	
2	tptc_a0_err	R/W	0h	
1	tpcc_a_mpint	R/W	0h	
0	tpcc_a_errint	R/W	0h	

### 5.2.6.3 RSS\_TPCC\_A\_ERRAGG\_STATUS Register (Offset = Ch) [Reset = 00XXXXXXh]

RSS\_TPCC\_A\_ERRAGG\_STATUS is shown in [Table 5-480](#).

Return to the [Summary Table](#).

**Table 5-480. RSS\_TPCC\_A\_ERRAGG\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	tptc_a0_read_access_error	R/W	0h	
24	tpcc_a_read_access_error	R/W	0h	
23-18	RESERVED	R/W	0h	
17	tptc_a0_write_access_error	R/W	0h	
16	tpcc_a_write_access_error	R/W	0h	
15-9	RESERVED	R/W	0h	
8	tpcc_a_parity_err	R/W	0h	
7-3	RESERVED	R/W	0h	
2	tptc_a0_err	R/W	0h	
1	tpcc_a_mpint	R/W	0h	
0	tpcc_a_errint	R/W	0h	

#### 5.2.6.4 RSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW Register (Offset = 10h) [Reset = 00XXXXXXh]

RSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW is shown in [Table 5-481](#).

Return to the [Summary Table](#).

**Table 5-481. RSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R/W	0h	
25	tptc_a0_read_access_error	R/W	0h	
24	tpcc_a_read_access_error	R/W	0h	
23-18	RESERVED	R/W	0h	
17	tptc_a0_write_access_error	R/W	0h	
16	tpcc_a_write_access_error	R/W	0h	
15-9	RESERVED	R/W	0h	
8	tpcc_a_parity_err	R/W	0h	
7-3	RESERVED	R/W	0h	
2	tptc_a0_err	R/W	0h	
1	tpcc_a_mpint	R/W	0h	
0	tpcc_a_errint	R/W	0h	

### 5.2.6.5 RSS\_TPCC\_A\_INTAGG\_MASK Register (Offset = 14h) [Reset = 0000XX00h]

RSS\_TPCC\_A\_INTAGG\_MASK is shown in [Table 5-482](#).

Return to the [Summary Table](#).

**Table 5-482. RSS\_TPCC\_A\_INTAGG\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	tpcc_a0	R/W	0h	Mask Interrupt from TPTC A0 to aggregated Interrupt RCSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	
7	tpcc_a_int6	R/W	0h	
6	tpcc_a_int5	R/W	0h	
5	tpcc_a_int4	R/W	0h	
4	tpcc_a_int3	R/W	0h	
3	tpcc_a_int2	R/W	0h	
2	tpcc_a_int1	R/W	0h	
1	tpcc_a_int0	R/W	0h	
0	tpcc_a_intg	R/W	0h	



### 5.2.6.6 RSS\_TPCC\_A\_INTAGG\_STATUS Register (Offset = 18h) [Reset = 0000XX00h]

RSS\_TPCC\_A\_INTAGG\_STATUS is shown in [Table 5-483](#).

Return to the [Summary Table](#).

**Table 5-483. RSS\_TPCC\_A\_INTAGG\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	tpcc_a0	R/W	0h	Status of Interrupt from TPTC A0. Set only if Interupt is unmasked in RCSS_TPCC_A_INTAGG_MASK Wrie 0x1 to clear this interrupt.
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	
7	tpcc_a_int6	R/W	0h	
6	tpcc_a_int5	R/W	0h	
5	tpcc_a_int4	R/W	0h	
4	tpcc_a_int3	R/W	0h	
3	tpcc_a_int2	R/W	0h	
2	tpcc_a_int1	R/W	0h	
1	tpcc_a_int0	R/W	0h	
0	tpcc_a_intg	R/W	0h	

### 5.2.6.7 RSS\_TPCC\_A\_INTAGG\_STATUS\_RAW Register (Offset = 1Ch) [Reset = 0000XX00h]

RSS\_TPCC\_A\_INTAGG\_STATUS\_RAW is shown in [Table 5-484](#).

Return to the [Summary Table](#).

**Table 5-484. RSS\_TPCC\_A\_INTAGG\_STATUS\_RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	tpcc_a0	R/W	0h	Raw Status of Interrupt from TPTC A0. Set irrespective if the Interrupt is masked or unmasked in RCSS_TPCC_A_INTAGG_MASK
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	
7	tpcc_a_int6	R/W	0h	
6	tpcc_a_int5	R/W	0h	
5	tpcc_a_int4	R/W	0h	
4	tpcc_a_int3	R/W	0h	
3	tpcc_a_int2	R/W	0h	
2	tpcc_a_int1	R/W	0h	
1	tpcc_a_int0	R/W	0h	
0	tpcc_a_intg	R/W	0h	

### 5.2.6.8 RSS\_TPCC\_MEMINIT\_START Register (Offset = 20h) [Reset = 00000000h]

RSS\_TPCC\_MEMINIT\_START is shown in [Table 5-485](#).

Return to the [Summary Table](#).

**Table 5-485. RSS\_TPCC\_MEMINIT\_START Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	tpcc_a_meminit_start	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.

### 5.2.6.9 RSS\_TPCC\_MEMINIT\_DONE Register (Offset = 24h) [Reset = 00000000h]

RSS\_TPCC\_MEMINIT\_DONE is shown in [Table 5-486](#).

Return to the [Summary Table](#).

**Table 5-486. RSS\_TPCC\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	tpcc_a_meminit_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.6.10 RSS\_TPCC\_MEMINIT\_STATUS Register (Offset = 28h) [Reset = 00000000h]

RSS\_TPCC\_MEMINIT\_STATUS is shown in [Table 5-487](#).

Return to the [Summary Table](#).

**Table 5-487. RSS\_TPCC\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	tpcc_a_meminit_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.11 TPTC\_DBS\_CFG Register (Offset = 2Ch) [Reset = 00000000h]

TPTC\_DBS\_CFG is shown in [Table 5-488](#).

Return to the [Summary Table](#).

**Table 5-488. TPTC\_DBS\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-2	tptc_a1	R/W	0h	Max Burst size tieoff value for TPTC A1
1-0	tptc_a0	R/W	0h	Max Burst size tieoff value for TPTC A0

### 5.2.6.12 RSS\_TPCC\_A\_PARITY\_CTRL Register (Offset = 30h) [Reset = 00000000h]

RSS\_TPCC\_A\_PARITY\_CTRL is shown in [Table 5-489](#).

Return to the [Summary Table](#).

**Table 5-489. RSS\_TPCC\_A\_PARITY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	parity_err_clr	R/W	0h	Write 0x1 to clear the Parit Error status for TPCC
1	parity_testen	R/W	0h	Enable Parity Test for TPCC. Write 0x1 : Parity Test is enabled on PARAM memory
0	parity_en	R/W	0h	Enable Parity for TPCC. Write 0x1 : Parity is enabled on PARAM memory

### 5.2.6.13 RSS\_TPCC\_A\_PARITY\_STATUS Register (Offset = 34h) [Reset = 000000XXh]

RSS\_TPCC\_A\_PARITY\_STATUS is shown in [Table 5-490](#).

Return to the [Summary Table](#).

**Table 5-490. RSS\_TPCC\_A\_PARITY\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	parity_addr	R	0h	TPCC Error Address at which Parity Error occurred
7-0	RESERVED	R	0h	



#### 5.2.6.14 RSS\_TPTC\_BOUNDARY\_CFG Register (Offset = 274h) [Reset = 00000012h]

RSS\_TPTC\_BOUNDARY\_CFG is shown in [Table 5-491](#).

Return to the [Summary Table](#).

**Table 5-491. RSS\_TPTC\_BOUNDARY\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-0	tptc_a0_size	R/W	12h	Writing 1'b1 will disable the CID-RID-SID reordering feature for the TPTC instance

### 5.2.6.15 RSS\_TPTC\_XID\_REORDER\_CFG Register (Offset = 278h) [Reset = 00000000h]

RSS\_TPTC\_XID\_REORDER\_CFG is shown in [Table 5-492](#).

Return to the [Summary Table](#).

**Table 5-492. RSS\_TPTC\_XID\_REORDER\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	tptc_a0_disable	R/W	0h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of TPTC Example: writing 6'd19 decides boundary to be $2^{19}$ i.e. 512 KB

### 5.2.6.16 DBG\_ACK\_CPU\_CTRL Register (Offset = 27Ch) [Reset = 00000000h]

DBG\_ACK\_CPU\_CTRL is shown in [Table 5-493](#).

Return to the [Summary Table](#).

**Table 5-493. DBG\_ACK\_CPU\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	sel	R/W	0h	Select the Processor Suspend that is used to Suspend the DSS Peripehrals 0:MSS CR5 1:RESERVED 2:RESERVED 3-7:RSS CR4

### 5.2.6.17 RSS\_ADCBUF\_PING\_MEMINIT Register (Offset = 280h) [Reset = 00000000h]

RSS\_ADCBUF\_PING\_MEMINIT is shown in [Table 5-494](#).

Return to the [Summary Table](#).

**Table 5-494. RSS\_ADCBUF\_PING\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	start	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.

### 5.2.6.18 RSS\_ADCBUF\_PING\_MEMINIT\_DONE Register (Offset = 284h) [Reset = 00000000h]

RSS\_ADCBUF\_PING\_MEMINIT\_DONE is shown in [Table 5-495](#).

Return to the [Summary Table](#).

**Table 5-495. RSS\_ADCBUF\_PING\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.6.19 RSS\_ADCBUF\_PING\_MEMINIT\_STATUS Register (Offset = 288h) [Reset = 00000000h]

RSS\_ADCBUF\_PING\_MEMINIT\_STATUS is shown in [Table 5-496](#).

Return to the [Summary Table](#).

**Table 5-496. RSS\_ADCBUF\_PING\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.20 RSS\_ADCBUF\_PONG\_MEMINIT Register (Offset = 28Ch) [Reset = 00000000h]

RSS\_ADCBUF\_PONG\_MEMINIT is shown in [Table 5-497](#).

Return to the [Summary Table](#).

**Table 5-497. RSS\_ADCBUF\_PONG\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	start	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.

### 5.2.6.21 RSS\_ADCBUF\_PONG\_MEMINIT\_DONE Register (Offset = 290h) [Reset = 00000000h]

RSS\_ADCBUF\_PONG\_MEMINIT\_DONE is shown in [Table 5-498](#).

Return to the [Summary Table](#).

**Table 5-498. RSS\_ADCBUF\_PONG\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.



### 5.2.6.22 RSS\_ADCBUF\_PONG\_MEMINIT\_STATUS Register (Offset = 294h) [Reset = 00000000h]

RSS\_ADCBUF\_PONG\_MEMINIT\_STATUS is shown in [Table 5-499](#).

Return to the [Summary Table](#).

**Table 5-499. RSS\_ADCBUF\_PONG\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.23 SOC\_TO\_BSS\_SW\_INT Register (Offset = 2C8h) [Reset = 00000000h]

SOC\_TO\_BSS\_SW\_INT is shown in [Table 5-500](#).

Return to the [Summary Table](#).

**Table 5-500. SOC\_TO\_BSS\_SW\_INT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	trig	R/W	0h	Write Pulse Bit field writing to each bit field <0-7>: 1'b1:triggers BSS_SW_INT_RSS_CTRL<0-7> to BSS

#### 5.2.6.24 RSS\_DBG\_ACK\_CTL0 Register (Offset = 2CCh) [Reset = 00000000h]

RSS\_DBG\_ACK\_CTL0 is shown in [Table 5-501](#).

Return to the [Summary Table](#).

**Table 5-501. RSS\_DBG\_ACK\_CTL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	frc	R/W	0h	emulation suspend signal control . Writing '111' would ungate the emulation suspend signal to the FRC

### 5.2.6.25 DMMSWINT1 Register (Offset = 2D0h) [Reset = 00XXXXXXh]

DMMSWINT1 is shown in [Table 5-502](#).

Return to the [Summary Table](#).

**Table 5-502. DMMSWINT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22	DMMCQWREN	R/W	0h	CQ Write Enable from DMM. 0 --> Write to CQ memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from CQ_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMCQPINPONSEL register.
21	DMMCQPINPONSEL	R/W	0h	CQ Ping Pong select for HIL Mode
20-19	RESERVED	R/W	0h	
18	DMMCPWREN	R/W	0h	Writing 1'b1: Enables DMM writes in to CP read registers 1'b0: Disables DMM writes to CP read registers
17	DMMADCBUFWREN	R/W	0h	ADC Buffer Write Enable from DMM. 0 --> Write to ADC BUF memory will happen from DFE and Ping-pong select will come from HW FSM (same as ADC Buffer ping-pong select). 1 --> Write to CQ memory will happen from ADCBUF_W slave port in DSS interconnect using DMM as master and Ping-pong select will come from DMMADCBUFPINPONSEL register.
16	DMMADCBUFPINPONSEL	R/W	0h	ADC Buffer Ping Pong select for HIL Mode
15-0	RESERVED	R/W	0h	

### 5.2.6.26 RSS\_SHARED\_MEM\_MEMINIT Register (Offset = 2D4h) [Reset = 00000000h]

RSS\_SHARED\_MEM\_MEMINIT is shown in [Table 5-503](#).

Return to the [Summary Table](#).

**Table 5-503. RSS\_SHARED\_MEM\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	start	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.

### 5.2.6.27 RSS\_SHARED\_MEM\_MEMINIT\_DONE Register (Offset = 2D8h) [Reset = 00000000h]

RSS\_SHARED\_MEM\_MEMINIT\_DONE is shown in [Table 5-504](#).

Return to the [Summary Table](#).

**Table 5-504. RSS\_SHARED\_MEM\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.6.28 RSS\_SHARED\_MEM\_MEMINIT\_STATUS Register (Offset = 2DCh) [Reset = 00000000h]

RSS\_SHARED\_MEM\_MEMINIT\_STATUS is shown in [Table 5-505](#).

Return to the [Summary Table](#).

**Table 5-505. RSS\_SHARED\_MEM\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.29 BSS\_CONTROL Register (Offset = 400h) [Reset = 0XXXX000h]

BSS\_CONTROL is shown in [Table 5-506](#).

Return to the [Summary Table](#).

**Table 5-506. BSS\_CONTROL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30-28	dss_l3_access	R/W	0h	writing 3'b111 allocates DSS_L3_BANKB1 256KB as TCM for RSS_CR4
27	RESERVED	R/W	0h	
26-24	halt	R/W	7h	writing 3'b000 unhalts BSS. This is expected to be written only once per processor reset cycle.
23-19	RESERVED	R/W	0h	
18-16	wfi_override	R/W	0h	writing 3'b111 overrides the WFI signal from CR4 and asserts it.
15-12	RESERVED	R/W	0h	
11-0	bootmode	R/W	0h	writing 12'h000 : selects the normal boot mode for CR4. 12'h111 : selects the FW dev mode for CR4 12'h222 : selects the orbit mode for CR4 12'h333 : selects the 14xx ROM swap mode



### 5.2.6.30 BSS\_TCM\_MEMINIT Register (Offset = 404h) [Reset = 00000000h]

BSS\_TCM\_MEMINIT is shown in [Table 5-507](#).

Return to the [Summary Table](#).

**Table 5-507. BSS\_TCM\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization.

### 5.2.6.31 BSS\_TCM\_MEMINIT\_DONE Register (Offset = 408h) [Reset = 00000000h]

BSS\_TCM\_MEMINIT\_DONE is shown in [Table 5-508](#).

Return to the [Summary Table](#).

**Table 5-508. BSS\_TCM\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.6.32 BSS\_TCM\_MEMINIT\_STATUS Register (Offset = 40Ch) [Reset = 00000000h]

BSS\_TCM\_MEMINIT\_STATUS is shown in [Table 5-509](#).

Return to the [Summary Table](#).

**Table 5-509. BSS\_TCM\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.33 BSS\_VIM\_MEMINIT Register (Offset = 410h) [Reset = 00000000h]

BSS\_VIM\_MEMINIT is shown in [Table 5-510](#).

Return to the [Summary Table](#).

**Table 5-510. BSS\_VIM\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization.

### 5.2.6.34 BSS\_VIM\_MEMINIT\_DONE Register (Offset = 414h) [Reset = 00000000h]

BSS\_VIM\_MEMINIT\_DONE is shown in [Table 5-511](#).

Return to the [Summary Table](#).

**Table 5-511. BSS\_VIM\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.6.35 BSS\_VIM\_MEMINIT\_STATUS Register (Offset = 418h) [Reset = 00000000h]

BSS\_VIM\_MEMINIT\_STATUS is shown in [Table 5-512](#).

Return to the [Summary Table](#).

**Table 5-512. BSS\_VIM\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.36 BSS\_DFE\_MEMINIT Register (Offset = 41Ch) [Reset = 00000000h]

BSS\_DFE\_MEMINIT is shown in [Table 5-513](#).

Return to the [Summary Table](#).

**Table 5-513. BSS\_DFE\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization.

### 5.2.6.37 BSS\_DFE\_MEMINIT\_DONE Register (Offset = 420h) [Reset = 00000000h]

BSS\_DFE\_MEMINIT\_DONE is shown in [Table 5-514](#).

Return to the [Summary Table](#).

**Table 5-514. BSS\_DFE\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.



### 5.2.6.38 BSS\_DFE\_MEMINIT\_STATUS Register (Offset = 424h) [Reset = 00000000h]

BSS\_DFE\_MEMINIT\_STATUS is shown in [Table 5-515](#).

Return to the [Summary Table](#).

**Table 5-515. BSS\_DFE\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.6.39 BSS\_RAMPGEN\_MEMINIT Register (Offset = 428h) [Reset = 00000000h]

BSS\_RAMPGEN\_MEMINIT is shown in [Table 5-516](#).

Return to the [Summary Table](#).

**Table 5-516. BSS\_RAMPGEN\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization.

### 5.2.6.40 BSS\_RAMPGEN\_MEMINIT\_DONE Register (Offset = 42Ch) [Reset = 00000000h]

BSS\_RAMPGEN\_MEMINIT\_DONE is shown in [Table 5-517](#).

Return to the [Summary Table](#).

**Table 5-517. BSS\_RAMPGEN\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

#### 5.2.6.41 BSS\_RAMPGEN\_MEMINIT\_STATUS Register (Offset = 430h) [Reset = 00000000h]

BSS\_RAMPGEN\_MEMINIT\_STATUS is shown in [Table 5-518](#).

Return to the [Summary Table](#).

**Table 5-518. BSS\_RAMPGEN\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

#### 5.2.6.42 BSS\_DSS\_L3\_STICKY Register (Offset = 434h) [Reset = 00000000h]

BSS\_DSS\_L3\_STICKY is shown in [Table 5-519](#).

Return to the [Summary Table](#).

**Table 5-519. BSS\_DSS\_L3\_STICKY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	sticky_enable	R/W	0h	writing 3'b111 make the BSS_CONTROL::DSS_L3_ACCESS_ENABLE sticky. Further writes to DSS_L3_ACCESS_ENABLE wont impact the register

### 5.2.6.43 BSS\_DSS\_L3\_ACCESS Register (Offset = 438h) [Reset = 00000000h]

BSS\_DSS\_L3\_ACCESS is shown in [Table 5-520](#).

Return to the [Summary Table](#).

**Table 5-520. BSS\_DSS\_L3\_ACCESS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	status	R	0h	reading 1'b0: DSS_L3_BANKD1 is not allocated to BSS_TCMA 1'b1: DSS_L3_BANKD1 is allocated to BSS_TCMA

#### 5.2.6.44 TESTPATTERNRX1ICFG Register (Offset = 800h) [Reset = 00010000h]

TESTPATTERNRX1ICFG is shown in [Table 5-521](#).

Return to the [Summary Table](#).

**Table 5-521. TESTPATTERNRX1ICFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

### 5.2.6.45 TESTPATTERNRX2ICFG Register (Offset = 804h) [Reset = 00010000h]

TESTPATTERNRX2ICFG is shown in [Table 5-522](#).

Return to the [Summary Table](#).

**Table 5-522. TESTPATTERNRX2ICFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 1.
15-0	TSTPATRX2IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 1.



### 5.2.6.46 TESTPATTERNRX3ICFG Register (Offset = 808h) [Reset = 00010000h]

TESTPATTERNRX3ICFG is shown in [Table 5-523](#).

Return to the [Summary Table](#).

**Table 5-523. TESTPATTERNRX3ICFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 2
15-0	TSTPATRX3IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 2

### 5.2.6.47 TESTPATTERNRX4ICFG Register (Offset = 80Ch) [Reset = 00010000h]

TESTPATTERNRX4ICFG is shown in [Table 5-524](#).

Return to the [Summary Table](#).

**Table 5-524. TESTPATTERNRX4ICFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4IINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in I channel Rx channel 3
15-0	TSTPATRX4IOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in I channel Rx channel 3

### 5.2.6.48 TESTPATTERNRX1QCFG Register (Offset = 810h) [Reset = 00010000h]

TESTPATTERNRX1QCFG is shown in [Table 5-525](#).

Return to the [Summary Table](#).

**Table 5-525. TESTPATTERNRX1QCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX1QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.
15-0	TSTPATRX1QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 0. In this register the naming convention for the 4 Rx channel indices are from 1 to 4 instead of 0 to 3.

### 5.2.6.49 TESTPATTERNRX2QCFG Register (Offset = 814h) [Reset = 00010000h]

TESTPATTERNRX2QCFG is shown in [Table 5-526](#).

Return to the [Summary Table](#).

**Table 5-526. TESTPATTERNRX2QCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX2QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 1.
15-0	TSTPATRX2QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 1.

### 5.2.6.50 TESTPATTERNRX3QCFG Register (Offset = 818h) [Reset = 00010000h]

TESTPATTERNRX3QCFG is shown in [Table 5-527](#).

Return to the [Summary Table](#).

**Table 5-527. TESTPATTERNRX3QCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX3QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 2
15-0	TSTPATRX3QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 2

### 5.2.6.51 TESTPATTERNRX4QCFG Register (Offset = 81Ch) [Reset = 00010000h]

TESTPATTERNRX4QCFG is shown in [Table 5-528](#).

Return to the [Summary Table](#).

**Table 5-528. TESTPATTERNRX4QCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TSTPATRX4QINCR	R/W	1h	Value to be added for each successive sample for the test pattern data in Q channel Rx channel 3
15-0	TSTPATRX4QOFFSET	R/W	0h	Offset value to be used for the first sample for the test pattern data in Q channel Rx channel 3

### 5.2.6.52 TESTPATTERNVLDCFG Register (Offset = 820h) [Reset = 00000008h]

TESTPATTERNVLDCFG is shown in [Table 5-529](#).

Return to the [Summary Table](#).

**Table 5-529. TESTPATTERNVLDCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	TSTPATGENEN	R/W	0h	Enable for test pattern generator. This is used to Mux with the functional data from BSS. 000 -->Disable, 111-->Enable, Others are reserved.
7-0	TSTPATVLDCNT	R/W	8h	Number of DSS Interconnect clocks (200 MHz) between successive samples for the test pattern gen.

### 5.2.6.53 ADCBUFCFG1 Register (Offset = 824h) [Reset = 00010004h]

ADCBUFCFG1 is shown in [Table 5-530](#).

Return to the [Summary Table](#).

**Table 5-530. ADCBUFCFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	ADCBUFPIPOSEL	R	1h	TI Internal Feature Ping-pong select value from ADC Buffer Packing logic. Even in SW override mode, this register will indicate the ping-pong select signal generated from the ADC Buffer Packing logic and not the override value.
15	ADCBUFCONTSTOPPL	R/W	0h	Stop Pulse for Continuous mode. The data capture will stop once this register is set. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode : Its a wspecial access type, write to this field will generate a pulse
14	ADCBUFCONTSTRTP	R/W	0h	Start Pulse for Continuous mode. The data capture will start from Address 0 once this register is set. All the other configurations like Enable, Sample Count are expected to be programmed before this pulse. Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode : Its a wspecial access type, write to this field will generate a pulse
13	ADCBUFCONTMODEEN	R/W	0h	Continuous mode enable for ADC Buffer. This is set when a fixed number of samples have to be stored in Ping/Pong and not depend on Chirp time-lines (Eg: Analog Lab characterization to stream out continuous data from DFE). Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode
12	ADCBUFWRITEMODE	R/W	0h	This needs to be programmed to 0x1 in AR16xx 0 --> Interleaved, 1 --> Non-interleaved
11	ADCBUFPIPOOVRVAL	R/W	0h	TI Internal Feature SW override value for ADC Buffer Ping Pong select
10	ADCBUFPIPOOVRCNT	R/W	0h	TI Internal Feature Override control for ADC Buffer Ping Pong select
9	RX3EN	R/W	0h	Enable for Rx3 write
8	RX2EN	R/W	0h	Enable for Rx2 write
7	RX1EN	R/W	0h	Enable for Rx1 write
6	RX0EN	R/W	0h	Enable for Rx0 write
5	ADCBUFIQSWAP	R/W	0h	0 --> I is stored in LSB and Q is stored in MSB 1 --> Q is stored in LSB and I is stored in MSB
4	ADCBUFRL2CHINTRL	R/W	0h	TI reserved field. Do not touch
3	ADCBUFRLMODECHSEL	R/W	0h	TI Internal Feature 0 -->I channel, 1 --> Q channel
2	ADCBUFREALONLYMODE	R/W	1h	0-->Complex Data mode, 1-->Real data mode
1	ADCBUFPIPOSELINV	R/W	0h	TI Internal Feature Inversion control for ADC Buffer Ping-pong select. By default ADC Buffer write starts with Pong write. By setting this bit to 1, it will start from Ping write after reset.
0	ADCBUFWRSOURCE	R/W	0h	TI Internal Feature Write source for ADC Buffer. 0 --> DFE, 1 --> VIN



### 5.2.6.54 ADCBUFCFG1\_EXTD Register (Offset = 828h) [Reset = 00000000h]

ADCBUFCFG1\_EXTD is shown in [Table 5-531](#).

Return to the [Summary Table](#).

**Table 5-531. ADCBUFCFG1\_EXTD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADCBUFINTGENDLY	R/W	0h	TI Internal Feature. No of clocks to delay the ping-pong switch and interrupt generation w.r.t ADC Valid fall pulse. This will enable dithering the DSP activity for successive ping-pong switch cycles. This will not delay the ping pong toggle which will happen immediately after ADC Valid fall.

### 5.2.6.55 ADCBUFCFG2 Register (Offset = 82Ch) [Reset = 0100XX00h]

ADCBUFCFG2 is shown in [Table 5-532](#).

Return to the [Summary Table](#).

**Table 5-532. ADCBUFCFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	ADCBUFADDRX1	R/W	100h	128 bit Address offset to be added to the internal address pointer for Rx1 writes in Non-interleaved mode.
15-11	RESERVED	R/W	0h	
10-0	ADCBUFADDRX0	R/W	0h	128 bit Address offset to be added to the internal address pointer for Rx0 writes in Non-interleaved mode.

### 5.2.6.56 ADCBUFCFG3 Register (Offset = 830h) [Reset = 0300XX00h]

ADCBUFCFG3 is shown in [Table 5-533](#).

Return to the [Summary Table](#).

**Table 5-533. ADCBUFCFG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	ADCBUFADDRX3	R/W	300h	128 bit Address offset to be added to the internal address pointer for Rx3 writes in Non-interleaved mode.
15-11	RESERVED	R/W	0h	
10-0	ADCBUFADDRX2	R/W	200h	128 bit Address offset to be added to the internal address pointer for Rx2 writes in Non-interleaved mode.

### 5.2.6.57 ADCBUF CFG4 Register (Offset = 834h) [Reset = XX000400h]

ADCBUF CFG4 is shown in [Table 5-534](#).

Return to the [Summary Table](#).

**Table 5-534. ADCBUF CFG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30	ADCBUF PNGSEL TGLDIS	R/W	0h	TI Internal Feature 0 --> Delay Interrupt Gen and Ping/Pong toggle together based on cfg_interrupt_gen_delay, 1 --> Delay only Interrupt Gen based on cfg_interrupt_gen_delay. But toggle Ping/Pong select signal as soon as the write is complete.
29-26	RESERVED	R/W	0h	
25-21	ADCBUF NUMCHR PPONG	R/W	0h	Number of chirps to be stored in Pong buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Ping
20-16	ADCBUF NUMCHR PPING	R/W	0h	Number of chirps to be stored in Ping buffer. This register should be programmed with one less than the actual number needed. This is used when data is written to Pong Memory. The value written to this field should be the same as that configured for Pong
15-0	ADCBUF SAMP CNT	R/W	400h	No of samples to store in each Ping and Pong register in continuous mode of ADC Buffer. In real only mode this refers to the number of real samples and in complex mode, this refers to number of complex samples. This refers to the number of samples per channel. This counter increments once for every new sample from DFE (as long as 1 or more channels are enabled). The max allowed value varies depending on other configurations (No of channels enabled and real/complex data). Continuous mode is expected to be only used for CZ and ADC Buffer Testpattern mode

### 5.2.6.58 ADCBUFINTGENDITHERDLY Register (Offset = 838h) [Reset = 00000000h]

ADCBUFINTGENDITHERDLY is shown in [Table 5-535](#).

Return to the [Summary Table](#).

**Table 5-535. ADCBUFINTGENDITHERDLY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADCBUFINTGENDITHERDLY	R/W	0h	TI Internal Feature. Additional dithering delay added on the Chirp Available interrupt

### 5.2.6.59 CBUFF\_FRAME\_START\_SEL Register (Offset = 83Ch) [Reset = 00000000h]

CBUFF\_FRAME\_START\_SEL is shown in [Table 5-536](#).

Return to the [Summary Table](#).

**Table 5-536. CBUFF\_FRAME\_START\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	sel	R/W	0h	writing: 1'b0: selects frame_start from DFE 1'b1: Selects frame_start from chirp_avail (adc capture complete)

### 5.2.6.60 CQCFG1 Register (Offset = C00h) [Reset = 4010000Xh]

CQCFG1 is shown in [Table 5-537](#).

Return to the [Summary Table](#).

**Table 5-537. CQCFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R/W	0h	
30-22	CQ2BASEADDR	R/W	100h	128-bit Address offset which indicates the start address for storing CQ0 (ADC/RxIF Saturation Detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
21-13	CQ1BASEADDR	R/W	80h	128-bit Address offset which indicates the start address for storing CQ0 (Signal Image Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
12-4	CQ0BASEADDR	R/W	0h	128-bit Address offset which indicates the start address for storing CQ0 (Wide Band Energy detection) from the start of CQ memory. This is not the byte address offset but 128 bit address offset
3	CQ96BITPACKEN	R/W	0h	This is used to pack the CQ data into only the LSB 96 bits of each row of the CQ memory. This can be used in 3 channel mode of LVDS where the ADC data and Chirp Params occupy only LSB 96 bits of each memory row.
2	RESERVED	R/W	0h	
1-0	CQDATAWIDTH	R/W	0h	This is used to appropriately pack the valid CQ data bits in appropriate bits in the CQ memory. 00, 01-->Raw 16, 10-->Raw 12, 11-->Raw14

### 5.2.6.61 CQCFG2 Register (Offset = C04h) [Reset = 0000XXXh]

CQCFG2 is shown in [Table 5-538](#).

Return to the [Summary Table](#).

**Table 5-538. CQCFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	CQ_CLK_GATE	R/W	0h	writing: 1'b0: ungates the clk to CQ logic 1'b1: Gates the clk to CQ logic
16	CQPIPOSELVAL	R/W	0h	Ping pong select override value for CQ Memory. 1 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to ping memory and write access from CQ_W/DFE write will be routed to pong memory. 0 --> Read access from Chirp Info Slave of DSS Interconnect will be routed to pong memory and write access from CQ_W/DFE write will be routed to ping memory.
15-13	RESERVED	R/W	0h	
12	CQPIPOSELCNT	R/W	0h	Ping pong select override control for CQ Memory. 0 --> Ping-pong select comes from HW FSM (same as the ping-pong select for ADC Buffer)/DMMCQPINPONSEL 1 --> Ping pong select for CQ memory is taken from SW register (CQPIPOSELVAL)
11-9	RESERVED	R/W	0h	
8	CQ2TESTMODEEN	R/W	0h	TI Internal Feaure Test Mode enable for CQ2 (ADC/RxIF Saturation). Once enabled, each 8 bit data is same as Addr+1.
7-5	RESERVED	R/W	0h	
4	CQ1TESTMODEEN	R/W	0h	TI Internal Feaure Test Mode enable for CQ1 (SI). Once enabled, each 16 bit data is same as [2*Addr+1 for the MSB 8 bits and Addr+1 for the LSB 8 bits.
3-1	RESERVED	R/W	0h	
0	CQ0TESTMODEEN	R/W	0h	TI Internal Feaure Test Mode enable for CQ0 (WBE). Once enabled, each 16 bit data is same as [2*Addr+1 for the MSB 8 bits and Addr+1 for the LSB 8 bits.



### 5.2.6.62 CPREG0 Register (Offset = C08h) [Reset = 00000000h]

CPREG0 is shown in [Table 5-539](#).

Return to the [Summary Table](#).

**Table 5-539. CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG0	R	0h	Chirp Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.63 CPREG1 Register (Offset = C0Ch) [Reset = 00000001h]

CPREG1 is shown in [Table 5-540](#).

Return to the [Summary Table](#).

**Table 5-540. CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG1	R	1h	Chirp Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.64 CPREG2 Register (Offset = C10h) [Reset = 00000002h]

CPREG2 is shown in [Table 5-541](#).

Return to the [Summary Table](#).

**Table 5-541. CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG2	R	2h	Chirp Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.65 CPREG3 Register (Offset = C14h) [Reset = 00000003h]

CPREG3 is shown in [Table 5-542](#).

Return to the [Summary Table](#).

**Table 5-542. CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG3	R	3h	Chirp Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.66 CPREG4 Register (Offset = C18h) [Reset = 00000000h]

CPREG4 is shown in [Table 5-543](#).

Return to the [Summary Table](#).

**Table 5-543. CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG4	R	0h	Chirp Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.67 CPREG5 Register (Offset = C1Ch) [Reset = 00000000h]

CPREG5 is shown in [Table 5-544](#).

Return to the [Summary Table](#).

**Table 5-544. CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG5	R	0h	Chirp Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.68 CPREG6 Register (Offset = C20h) [Reset = 00000000h]

CPREG6 is shown in [Table 5-545](#).

Return to the [Summary Table](#).

**Table 5-545. CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG6	R	0h	Chirp Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.69 CPREG7 Register (Offset = C24h) [Reset = 00000000h]

CPREG7 is shown in [Table 5-546](#).

Return to the [Summary Table](#).

**Table 5-546. CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG7	R	0h	Chirp Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.70 CPREG8 Register (Offset = C28h) [Reset = 00000000h]

CPREG8 is shown in [Table 5-547](#).

Return to the [Summary Table](#).

**Table 5-547. CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG8	R	0h	Chirp Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.71 CPREG9 Register (Offset = C2Ch) [Reset = 00000000h]

CPREG9 is shown in [Table 5-548](#).

Return to the [Summary Table](#).

**Table 5-548. CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG9	R	0h	Chirp Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.72 CPREG10 Register (Offset = C30h) [Reset = 00000000h]

CPREG10 is shown in [Table 5-549](#).

Return to the [Summary Table](#).

**Table 5-549. CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG10	R	0h	Chirp Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.73 CPREG11 Register (Offset = C34h) [Reset = 00000000h]

CPREG11 is shown in [Table 5-550](#).

Return to the [Summary Table](#).

**Table 5-550. CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG11	R	0h	Chirp Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

#### 5.2.6.74 CPREG12 Register (Offset = C38h) [Reset = 00000000h]

CPREG12 is shown in [Table 5-551](#).

Return to the [Summary Table](#).

**Table 5-551. CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG12	R	0h	Chirp Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.75 CPREG13 Register (Offset = C3Ch) [Reset = 00000000h]

CPREG13 is shown in [Table 5-552](#).

Return to the [Summary Table](#).

**Table 5-552. CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG13	R	0h	Chirp Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.76 CPREG14 Register (Offset = C40h) [Reset = 00000000h]

CPREG14 is shown in [Table 5-553](#).

Return to the [Summary Table](#).

**Table 5-553. CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG14	R	0h	Chirp Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.77 CPREG15 Register (Offset = C44h) [Reset = 00000000h]

CPREG15 is shown in [Table 5-554](#).

Return to the [Summary Table](#).

**Table 5-554. CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPREG15	R	0h	Chirp Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.78 CH0CPREG0 Register (Offset = C48h) [Reset = 00000000h]

CH0CPREG0 is shown in [Table 5-555](#).

Return to the [Summary Table](#).

**Table 5-555. CH0CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG0	R	0h	Multi Chirp 0 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.79 CH0CPREG1 Register (Offset = C4Ch) [Reset = 00000001h]

CH0CPREG1 is shown in [Table 5-556](#).

Return to the [Summary Table](#).

**Table 5-556. CH0CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG1	R	1h	Multi Chirp 0 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.80 CH0CPREG2 Register (Offset = C50h) [Reset = 00000002h]

CH0CPREG2 is shown in [Table 5-557](#).

Return to the [Summary Table](#).

**Table 5-557. CH0CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG2	R	2h	Multi Chirp 0 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.81 CH0CPREG3 Register (Offset = C54h) [Reset = 00000003h]

CH0CPREG3 is shown in [Table 5-558](#).

Return to the [Summary Table](#).

**Table 5-558. CH0CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG3	R	3h	Multi Chirp 0 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.82 CH0CPREG4 Register (Offset = C58h) [Reset = 00000000h]

CH0CPREG4 is shown in [Table 5-559](#).

Return to the [Summary Table](#).

**Table 5-559. CH0CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG4	R	0h	Multi Chirp 0 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.83 CH0CPREG5 Register (Offset = C5Ch) [Reset = 00000000h]

CH0CPREG5 is shown in [Table 5-560](#).

Return to the [Summary Table](#).

**Table 5-560. CH0CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG5	R	0h	Multi Chirp 0 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

#### 5.2.6.84 CH0CPREG6 Register (Offset = C60h) [Reset = 00000000h]

CH0CPREG6 is shown in [Table 5-561](#).

Return to the [Summary Table](#).

**Table 5-561. CH0CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG6	R	0h	Multi Chirp 0 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.85 CH0CPREG7 Register (Offset = C64h) [Reset = 00000000h]

CH0CPREG7 is shown in [Table 5-562](#).

Return to the [Summary Table](#).

**Table 5-562. CH0CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG7	R	0h	Multi Chirp 0 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.86 CH0CPREG8 Register (Offset = C68h) [Reset = 00000000h]

CH0CPREG8 is shown in [Table 5-563](#).

Return to the [Summary Table](#).

**Table 5-563. CH0CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG8	R	0h	Multi Chirp 0 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.87 CH0CPREG9 Register (Offset = C6Ch) [Reset = 00000000h]

CH0CPREG9 is shown in [Table 5-564](#).

Return to the [Summary Table](#).

**Table 5-564. CH0CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG9	R	0h	Multi Chirp 0 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.88 CH0CPREG10 Register (Offset = C70h) [Reset = 00000000h]

CH0CPREG10 is shown in [Table 5-565](#).

Return to the [Summary Table](#).

**Table 5-565. CH0CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG10	R	0h	Multi Chirp 0 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.89 CH0CPREG11 Register (Offset = C74h) [Reset = 00000000h]

CH0CPREG11 is shown in [Table 5-566](#).

Return to the [Summary Table](#).

**Table 5-566. CH0CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG11	R	0h	Multi Chirp 0 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.90 CH0CPREG12 Register (Offset = C78h) [Reset = 00000000h]

CH0CPREG12 is shown in [Table 5-567](#).

Return to the [Summary Table](#).

**Table 5-567. CH0CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG12	R	0h	Multi Chirp 0 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.91 CH0CPREG13 Register (Offset = C7Ch) [Reset = 00000000h]

CH0CPREG13 is shown in [Table 5-568](#).

Return to the [Summary Table](#).

**Table 5-568. CH0CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG13	R	0h	Multi Chirp 0 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.92 CH0CPREG14 Register (Offset = C80h) [Reset = 00000000h]

CH0CPREG14 is shown in [Table 5-569](#).

Return to the [Summary Table](#).

**Table 5-569. CH0CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG14	R	0h	Multi Chirp 0 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.93 CH0CPREG15 Register (Offset = C84h) [Reset = 00000000h]

CH0CPREG15 is shown in [Table 5-570](#).

Return to the [Summary Table](#).

**Table 5-570. CH0CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH0CPREG15	R	0h	Multi Chirp 0 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.94 CH1CPREG0 Register (Offset = C88h) [Reset = 00000000h]

CH1CPREG0 is shown in [Table 5-571](#).

Return to the [Summary Table](#).

**Table 5-571. CH1CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG0	R	0h	Multi Chirp 1 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.95 CH1CPREG1 Register (Offset = C8Ch) [Reset = 00000001h]

CH1CPREG1 is shown in [Table 5-572](#).

Return to the [Summary Table](#).

**Table 5-572. CH1CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG1	R	1h	Multi Chirp 1 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.96 CH1CPREG2 Register (Offset = C90h) [Reset = 00000002h]

CH1CPREG2 is shown in [Table 5-573](#).

Return to the [Summary Table](#).

**Table 5-573. CH1CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG2	R	2h	Multi Chirp 1 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.97 CH1CPREG3 Register (Offset = C94h) [Reset = 00000003h]

CH1CPREG3 is shown in [Table 5-574](#).

Return to the [Summary Table](#).

**Table 5-574. CH1CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG3	R	3h	Multi Chirp 1 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.98 CH1CPREG4 Register (Offset = C98h) [Reset = 00000000h]

CH1CPREG4 is shown in [Table 5-575](#).

Return to the [Summary Table](#).

**Table 5-575. CH1CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG4	R	0h	Multi Chirp 1 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.99 CH1CPREG5 Register (Offset = C9Ch) [Reset = 00000000h]

CH1CPREG5 is shown in [Table 5-576](#).

Return to the [Summary Table](#).

**Table 5-576. CH1CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG5	R	0h	Multi Chirp 1 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.100 CH1CPREG6 Register (Offset = CA0h) [Reset = 00000000h]

CH1CPREG6 is shown in [Table 5-577](#).

Return to the [Summary Table](#).

**Table 5-577. CH1CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG6	R	0h	Multi Chirp 1 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.101 CH1CPREG7 Register (Offset = CA4h) [Reset = 00000000h]

CH1CPREG7 is shown in [Table 5-578](#).

Return to the [Summary Table](#).

**Table 5-578. CH1CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG7	R	0h	Multi Chirp 1 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.102 CH1CPREG8 Register (Offset = CA8h) [Reset = 00000000h]

CH1CPREG8 is shown in [Table 5-579](#).

Return to the [Summary Table](#).

**Table 5-579. CH1CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG8	R	0h	Multi Chirp 1 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.103 CH1CPREG9 Register (Offset = CACH) [Reset = 00000000h]

CH1CPREG9 is shown in [Table 5-580](#).

Return to the [Summary Table](#).

**Table 5-580. CH1CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG9	R	0h	Multi Chirp 1 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.104 CH1CPREG10 Register (Offset = CB0h) [Reset = 00000000h]

CH1CPREG10 is shown in [Table 5-581](#).

Return to the [Summary Table](#).

**Table 5-581. CH1CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG10	R	0h	Multi Chirp 1 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.105 CH1CPREG11 Register (Offset = CB4h) [Reset = 00000000h]

CH1CPREG11 is shown in [Table 5-582](#).

Return to the [Summary Table](#).

**Table 5-582. CH1CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG11	R	0h	Multi Chirp 1 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.106 CH1CPREG12 Register (Offset = CB8h) [Reset = 00000000h]

CH1CPREG12 is shown in [Table 5-583](#).

Return to the [Summary Table](#).

**Table 5-583. CH1CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG12	R	0h	Multi Chirp 1 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.107 CH1CPREG13 Register (Offset = CBCh) [Reset = 00000000h]

CH1CPREG13 is shown in [Table 5-584](#).

Return to the [Summary Table](#).

**Table 5-584. CH1CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG13	R	0h	Multi Chirp 1 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.108 CH1CPREG14 Register (Offset = CC0h) [Reset = 00000000h]

CH1CPREG14 is shown in [Table 5-585](#).

Return to the [Summary Table](#).

**Table 5-585. CH1CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG14	R	0h	Multi Chirp 1 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.109 CH1CPREG15 Register (Offset = CC4h) [Reset = 00000000h]

CH1CPREG15 is shown in [Table 5-586](#).

Return to the [Summary Table](#).

**Table 5-586. CH1CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH1CPREG15	R	0h	Multi Chirp 1 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.110 CH2CPREG0 Register (Offset = CC8h) [Reset = 00000000h]

CH2CPREG0 is shown in [Table 5-587](#).

Return to the [Summary Table](#).

**Table 5-587. CH2CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG0	R	0h	Multi Chirp 2 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.111 CH2CPREG1 Register (Offset = CCCh) [Reset = 00000001h]

CH2CPREG1 is shown in [Table 5-588](#).

Return to the [Summary Table](#).

**Table 5-588. CH2CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG1	R	1h	Multi Chirp 2 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.112 CH2CPREG2 Register (Offset = CD0h) [Reset = 00000002h]

CH2CPREG2 is shown in [Table 5-589](#).

Return to the [Summary Table](#).

**Table 5-589. CH2CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG2	R	2h	Multi Chirp 2 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.113 CH2CPREG3 Register (Offset = CD4h) [Reset = 00000003h]

CH2CPREG3 is shown in [Table 5-590](#).

Return to the [Summary Table](#).

**Table 5-590. CH2CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG3	R	3h	Multi Chirp 2 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.114 CH2CPREG4 Register (Offset = CD8h) [Reset = 00000000h]

CH2CPREG4 is shown in [Table 5-591](#).

Return to the [Summary Table](#).

**Table 5-591. CH2CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG4	R	0h	Multi Chirp 2 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.115 CH2CPREG5 Register (Offset = CDCh) [Reset = 00000000h]

CH2CPREG5 is shown in [Table 5-592](#).

Return to the [Summary Table](#).

**Table 5-592. CH2CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG5	R	0h	Multi Chirp 2 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.116 CH2CPREG6 Register (Offset = CE0h) [Reset = 00000000h]

CH2CPREG6 is shown in [Table 5-593](#).

Return to the [Summary Table](#).

**Table 5-593. CH2CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG6	R	0h	Multi Chirp 2 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.117 CH2CPREG7 Register (Offset = CE4h) [Reset = 00000000h]

CH2CPREG7 is shown in [Table 5-594](#).

Return to the [Summary Table](#).

**Table 5-594. CH2CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG7	R	0h	Multi Chirp 2 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.118 CH2CPREG8 Register (Offset = CE8h) [Reset = 00000000h]

CH2CPREG8 is shown in [Table 5-595](#).

Return to the [Summary Table](#).

**Table 5-595. CH2CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG8	R	0h	Multi Chirp 2 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.119 CH2CPREG9 Register (Offset = CECh) [Reset = 00000000h]

CH2CPREG9 is shown in [Table 5-596](#).

Return to the [Summary Table](#).

**Table 5-596. CH2CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG9	R	0h	Multi Chirp 2 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.120 CH2CPREG10 Register (Offset = CF0h) [Reset = 00000000h]

CH2CPREG10 is shown in [Table 5-597](#).

Return to the [Summary Table](#).

**Table 5-597. CH2CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG10	R	0h	Multi Chirp 2 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.121 CH2CPREG11 Register (Offset = CF4h) [Reset = 00000000h]

CH2CPREG11 is shown in [Table 5-598](#).

Return to the [Summary Table](#).

**Table 5-598. CH2CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG11	R	0h	Multi Chirp 2 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.122 CH2CPREG12 Register (Offset = CF8h) [Reset = 00000000h]

CH2CPREG12 is shown in [Table 5-599](#).

Return to the [Summary Table](#).

**Table 5-599. CH2CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG12	R	0h	Multi Chirp 2 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.123 CH2CPREG13 Register (Offset = CFCh) [Reset = 00000000h]

CH2CPREG13 is shown in [Table 5-600](#).

Return to the [Summary Table](#).

**Table 5-600. CH2CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG13	R	0h	Multi Chirp 2 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.124 CH2CPREG14 Register (Offset = D00h) [Reset = 00000000h]

CH2CPREG14 is shown in [Table 5-601](#).

Return to the [Summary Table](#).

**Table 5-601. CH2CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG14	R	0h	Multi Chirp 2 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.125 CH2CPREG15 Register (Offset = D04h) [Reset = 00000000h]

CH2CPREG15 is shown in [Table 5-602](#).

Return to the [Summary Table](#).

**Table 5-602. CH2CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH2CPREG15	R	0h	Multi Chirp 2 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.126 CH3CPREG0 Register (Offset = D08h) [Reset = 00000000h]

CH3CPREG0 is shown in [Table 5-603](#).

Return to the [Summary Table](#).

**Table 5-603. CH3CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG0	R	0h	Multi Chirp 3 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.127 CH3CPREG1 Register (Offset = D0Ch) [Reset = 00000001h]

CH3CPREG1 is shown in [Table 5-604](#).

Return to the [Summary Table](#).

**Table 5-604. CH3CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG1	R	1h	Multi Chirp 3 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.128 CH3CPREG2 Register (Offset = D10h) [Reset = 00000002h]

CH3CPREG2 is shown in [Table 5-605](#).

Return to the [Summary Table](#).

**Table 5-605. CH3CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG2	R	2h	Multi Chirp 3 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.129 CH3CPREG3 Register (Offset = D14h) [Reset = 00000003h]

CH3CPREG3 is shown in [Table 5-606](#).

Return to the [Summary Table](#).

**Table 5-606. CH3CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG3	R	3h	Multi Chirp 3 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.130 CH3CPREG4 Register (Offset = D18h) [Reset = 00000000h]

CH3CPREG4 is shown in [Table 5-607](#).

Return to the [Summary Table](#).

**Table 5-607. CH3CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG4	R	0h	Multi Chirp 3 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.131 CH3CPREG5 Register (Offset = D1Ch) [Reset = 00000000h]

CH3CPREG5 is shown in [Table 5-608](#).

Return to the [Summary Table](#).

**Table 5-608. CH3CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG5	R	0h	Multi Chirp 3 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.132 CH3CPREG6 Register (Offset = D20h) [Reset = 00000000h]

CH3CPREG6 is shown in [Table 5-609](#).

Return to the [Summary Table](#).

**Table 5-609. CH3CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG6	R	0h	Multi Chirp 3 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.133 CH3CPREG7 Register (Offset = D24h) [Reset = 00000000h]

CH3CPREG7 is shown in [Table 5-610](#).

Return to the [Summary Table](#).

**Table 5-610. CH3CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG7	R	0h	Multi Chirp 3 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.134 CH3CPREG8 Register (Offset = D28h) [Reset = 00000000h]

CH3CPREG8 is shown in [Table 5-611](#).

Return to the [Summary Table](#).

**Table 5-611. CH3CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG8	R	0h	Multi Chirp 3 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.135 CH3CPREG9 Register (Offset = D2Ch) [Reset = 00000000h]

CH3CPREG9 is shown in [Table 5-612](#).

Return to the [Summary Table](#).

**Table 5-612. CH3CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG9	R	0h	Multi Chirp 3 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.136 CH3CPREG10 Register (Offset = D30h) [Reset = 00000000h]

CH3CPREG10 is shown in [Table 5-613](#).

Return to the [Summary Table](#).

**Table 5-613. CH3CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG10	R	0h	Multi Chirp 3 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.137 CH3CPREG11 Register (Offset = D34h) [Reset = 00000000h]

CH3CPREG11 is shown in [Table 5-614](#).

Return to the [Summary Table](#).

**Table 5-614. CH3CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG11	R	0h	Multi Chirp 3 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.138 CH3CPREG12 Register (Offset = D38h) [Reset = 00000000h]

CH3CPREG12 is shown in [Table 5-615](#).

Return to the [Summary Table](#).

**Table 5-615. CH3CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG12	R	0h	Multi Chirp 3 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.139 CH3CPREG13 Register (Offset = D3Ch) [Reset = 00000000h]

CH3CPREG13 is shown in [Table 5-616](#).

Return to the [Summary Table](#).

**Table 5-616. CH3CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG13	R	0h	Multi Chirp 3 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.140 CH3CPREG14 Register (Offset = D40h) [Reset = 00000000h]

CH3CPREG14 is shown in [Table 5-617](#).

Return to the [Summary Table](#).

**Table 5-617. CH3CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG14	R	0h	Multi Chirp 3 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.141 CH3CPREG15 Register (Offset = D44h) [Reset = 00000000h]

CH3CPREG15 is shown in [Table 5-618](#).

Return to the [Summary Table](#).

**Table 5-618. CH3CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH3CPREG15	R	0h	Multi Chirp 3 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.142 CH4CPREG0 Register (Offset = D48h) [Reset = 00000000h]

CH4CPREG0 is shown in [Table 5-619](#).

Return to the [Summary Table](#).

**Table 5-619. CH4CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG0	R	0h	Multi Chirp 4 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.143 CH4CPREG1 Register (Offset = D4Ch) [Reset = 00000001h]

CH4CPREG1 is shown in [Table 5-620](#).

Return to the [Summary Table](#).

**Table 5-620. CH4CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG1	R	1h	Multi Chirp 4 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.144 CH4CPREG2 Register (Offset = D50h) [Reset = 00000002h]

CH4CPREG2 is shown in [Table 5-621](#).

Return to the [Summary Table](#).

**Table 5-621. CH4CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG2	R	2h	Multi Chirp 4 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.145 CH4CPREG3 Register (Offset = D54h) [Reset = 00000003h]

CH4CPREG3 is shown in [Table 5-622](#).

Return to the [Summary Table](#).

**Table 5-622. CH4CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG3	R	3h	Multi Chirp 4 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.146 CH4CPREG4 Register (Offset = D58h) [Reset = 00000000h]

CH4CPREG4 is shown in [Table 5-623](#).

Return to the [Summary Table](#).

**Table 5-623. CH4CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG4	R	0h	Multi Chirp 4 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.147 CH4CPREG5 Register (Offset = D5Ch) [Reset = 00000000h]

CH4CPREG5 is shown in [Table 5-624](#).

Return to the [Summary Table](#).

**Table 5-624. CH4CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG5	R	0h	Multi Chirp 4 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.148 CH4CPREG6 Register (Offset = D60h) [Reset = 00000000h]

CH4CPREG6 is shown in [Table 5-625](#).

Return to the [Summary Table](#).

**Table 5-625. CH4CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG6	R	0h	Multi Chirp 4 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.149 CH4CPREG7 Register (Offset = D64h) [Reset = 00000000h]

CH4CPREG7 is shown in [Table 5-626](#).

Return to the [Summary Table](#).

**Table 5-626. CH4CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG7	R	0h	Multi Chirp 4 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.150 CH4CPREG8 Register (Offset = D68h) [Reset = 00000000h]

CH4CPREG8 is shown in [Table 5-627](#).

Return to the [Summary Table](#).

**Table 5-627. CH4CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG8	R	0h	Multi Chirp 4 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.151 CH4CPREG9 Register (Offset = D6Ch) [Reset = 00000000h]

CH4CPREG9 is shown in [Table 5-628](#).

Return to the [Summary Table](#).

**Table 5-628. CH4CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG9	R	0h	Multi Chirp 4 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.152 CH4CPREG10 Register (Offset = D70h) [Reset = 00000000h]

CH4CPREG10 is shown in [Table 5-629](#).

Return to the [Summary Table](#).

**Table 5-629. CH4CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG10	R	0h	Multi Chirp 4 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.153 CH4CPREG11 Register (Offset = D74h) [Reset = 00000000h]

CH4CPREG11 is shown in [Table 5-630](#).

Return to the [Summary Table](#).

**Table 5-630. CH4CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG11	R	0h	Multi Chirp 4 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.154 CH4CPREG12 Register (Offset = D78h) [Reset = 00000000h]

CH4CPREG12 is shown in [Table 5-631](#).

Return to the [Summary Table](#).

**Table 5-631. CH4CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG12	R	0h	Multi Chirp 4 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.155 CH4CPREG13 Register (Offset = D7Ch) [Reset = 00000000h]

CH4CPREG13 is shown in [Table 5-632](#).

Return to the [Summary Table](#).

**Table 5-632. CH4CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG13	R	0h	Multi Chirp 4 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.156 CH4CPREG14 Register (Offset = D80h) [Reset = 00000000h]

CH4CPREG14 is shown in [Table 5-633](#).

Return to the [Summary Table](#).

**Table 5-633. CH4CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG14	R	0h	Multi Chirp 4 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.157 CH4CPREG15 Register (Offset = D84h) [Reset = 00000000h]

CH4CPREG15 is shown in [Table 5-634](#).

Return to the [Summary Table](#).

**Table 5-634. CH4CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH4CPREG15	R	0h	Multi Chirp 4 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.158 CH5CPREG0 Register (Offset = D88h) [Reset = 00000000h]

CH5CPREG0 is shown in [Table 5-635](#).

Return to the [Summary Table](#).

**Table 5-635. CH5CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG0	R	0h	Multi Chirp 5 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.159 CH5CPREG1 Register (Offset = D8Ch) [Reset = 00000001h]

CH5CPREG1 is shown in [Table 5-636](#).

Return to the [Summary Table](#).

**Table 5-636. CH5CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG1	R	1h	Multi Chirp 5 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.160 CH5CPREG2 Register (Offset = D90h) [Reset = 00000002h]

CH5CPREG2 is shown in [Table 5-637](#).

Return to the [Summary Table](#).

**Table 5-637. CH5CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG2	R	2h	Multi Chirp 5 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.161 CH5CPREG3 Register (Offset = D94h) [Reset = 00000003h]

CH5CPREG3 is shown in [Table 5-638](#).

Return to the [Summary Table](#).

**Table 5-638. CH5CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG3	R	3h	Multi Chirp 5 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.162 CH5CPREG4 Register (Offset = D98h) [Reset = 00000000h]

CH5CPREG4 is shown in [Table 5-639](#).

Return to the [Summary Table](#).

**Table 5-639. CH5CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG4	R	0h	Multi Chirp 5 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.163 CH5CPREG5 Register (Offset = D9Ch) [Reset = 00000000h]

CH5CPREG5 is shown in [Table 5-640](#).

Return to the [Summary Table](#).

**Table 5-640. CH5CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG5	R	0h	Multi Chirp 5 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.164 CH5CPREG6 Register (Offset = DA0h) [Reset = 00000000h]

CH5CPREG6 is shown in [Table 5-641](#).

Return to the [Summary Table](#).

**Table 5-641. CH5CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG6	R	0h	Multi Chirp 5 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.165 CH5CPREG7 Register (Offset = DA4h) [Reset = 00000000h]

CH5CPREG7 is shown in [Table 5-642](#).

Return to the [Summary Table](#).

**Table 5-642. CH5CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG7	R	0h	Multi Chirp 5 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.166 CH5CPREG8 Register (Offset = DA8h) [Reset = 00000000h]

CH5CPREG8 is shown in [Table 5-643](#).

Return to the [Summary Table](#).

**Table 5-643. CH5CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG8	R	0h	Multi Chirp 5 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.167 CH5CPREG9 Register (Offset = DACH) [Reset = 00000000h]

CH5CPREG9 is shown in [Table 5-644](#).

Return to the [Summary Table](#).

**Table 5-644. CH5CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG9	R	0h	Multi Chirp 5 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.168 CH5CPREG10 Register (Offset = DB0h) [Reset = 00000000h]

CH5CPREG10 is shown in [Table 5-645](#).

Return to the [Summary Table](#).

**Table 5-645. CH5CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG10	R	0h	Multi Chirp 5 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.169 CH5CPREG11 Register (Offset = DB4h) [Reset = 00000000h]

CH5CPREG11 is shown in [Table 5-646](#).

Return to the [Summary Table](#).

**Table 5-646. CH5CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG11	R	0h	Multi Chirp 5 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.170 CH5CPREG12 Register (Offset = DB8h) [Reset = 00000000h]

CH5CPREG12 is shown in [Table 5-647](#).

Return to the [Summary Table](#).

**Table 5-647. CH5CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG12	R	0h	Multi Chirp 5 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.171 CH5CPREG13 Register (Offset = DBCh) [Reset = 00000000h]

CH5CPREG13 is shown in [Table 5-648](#).

Return to the [Summary Table](#).

**Table 5-648. CH5CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG13	R	0h	Multi Chirp 5 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.172 CH5CPREG14 Register (Offset = DC0h) [Reset = 00000000h]

CH5CPREG14 is shown in [Table 5-649](#).

Return to the [Summary Table](#).

**Table 5-649. CH5CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG14	R	0h	Multi Chirp 5 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.173 CH5CPREG15 Register (Offset = DC4h) [Reset = 00000000h]

CH5CPREG15 is shown in [Table 5-650](#).

Return to the [Summary Table](#).

**Table 5-650. CH5CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH5CPREG15	R	0h	Multi Chirp 5 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.174 CH6CPREG0 Register (Offset = DC8h) [Reset = 00000000h]

CH6CPREG0 is shown in [Table 5-651](#).

Return to the [Summary Table](#).

**Table 5-651. CH6CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG0	R	0h	Multi Chirp 6 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.175 CH6CPREG1 Register (Offset = DCCh) [Reset = 00000001h]

CH6CPREG1 is shown in [Table 5-652](#).

Return to the [Summary Table](#).

**Table 5-652. CH6CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG1	R	1h	Multi Chirp 6 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.176 CH6CPREG2 Register (Offset = DD0h) [Reset = 00000002h]

CH6CPREG2 is shown in [Table 5-653](#).

Return to the [Summary Table](#).

**Table 5-653. CH6CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG2	R	2h	Multi Chirp 6 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.177 CH6CPREG3 Register (Offset = DD4h) [Reset = 00000003h]

CH6CPREG3 is shown in [Table 5-654](#).

Return to the [Summary Table](#).

**Table 5-654. CH6CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG3	R	3h	Multi Chirp 6 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.178 CH6CPREG4 Register (Offset = DD8h) [Reset = 00000000h]

CH6CPREG4 is shown in [Table 5-655](#).

Return to the [Summary Table](#).

**Table 5-655. CH6CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG4	R	0h	Multi Chirp 6 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.179 CH6CPREG5 Register (Offset = DDCh) [Reset = 00000000h]

CH6CPREG5 is shown in [Table 5-656](#).

Return to the [Summary Table](#).

**Table 5-656. CH6CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG5	R	0h	Multi Chirp 6 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.180 CH6CPREG6 Register (Offset = DE0h) [Reset = 00000000h]

CH6CPREG6 is shown in [Table 5-657](#).

Return to the [Summary Table](#).

**Table 5-657. CH6CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG6	R	0h	Multi Chirp 6 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.181 CH6CPREG7 Register (Offset = DE4h) [Reset = 00000000h]

CH6CPREG7 is shown in [Table 5-658](#).

Return to the [Summary Table](#).

**Table 5-658. CH6CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG7	R	0h	Multi Chirp 6 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.182 CH6CPREG8 Register (Offset = DE8h) [Reset = 00000000h]

CH6CPREG8 is shown in [Table 5-659](#).

Return to the [Summary Table](#).

**Table 5-659. CH6CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG8	R	0h	Multi Chirp 6 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.183 CH6CPREG9 Register (Offset = DECh) [Reset = 00000000h]

CH6CPREG9 is shown in [Table 5-660](#).

Return to the [Summary Table](#).

**Table 5-660. CH6CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG9	R	0h	Multi Chirp 6 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.184 CH6CPREG10 Register (Offset = DF0h) [Reset = 00000000h]

CH6CPREG10 is shown in [Table 5-661](#).

Return to the [Summary Table](#).

**Table 5-661. CH6CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG10	R	0h	Multi Chirp 6 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.185 CH6CPREG11 Register (Offset = DF4h) [Reset = 00000000h]

CH6CPREG11 is shown in [Table 5-662](#).

Return to the [Summary Table](#).

**Table 5-662. CH6CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG11	R	0h	Multi Chirp 6 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.186 CH6CPREG12 Register (Offset = DF8h) [Reset = 00000000h]

CH6CPREG12 is shown in [Table 5-663](#).

Return to the [Summary Table](#).

**Table 5-663. CH6CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG12	R	0h	Multi Chirp 6 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.187 CH6CPREG13 Register (Offset = DFCh) [Reset = 00000000h]

CH6CPREG13 is shown in [Table 5-664](#).

Return to the [Summary Table](#).

**Table 5-664. CH6CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG13	R	0h	Multi Chirp 6 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.188 CH6CPREG14 Register (Offset = E00h) [Reset = 00000000h]

CH6CPREG14 is shown in [Table 5-665](#).

Return to the [Summary Table](#).

**Table 5-665. CH6CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG14	R	0h	Multi Chirp 6 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.189 CH6CPREG15 Register (Offset = E04h) [Reset = 00000000h]

CH6CPREG15 is shown in [Table 5-666](#).

Return to the [Summary Table](#).

**Table 5-666. CH6CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH6CPREG15	R	0h	Multi Chirp 6 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.190 CH7CPREG0 Register (Offset = E08h) [Reset = 00000000h]

CH7CPREG0 is shown in [Table 5-667](#).

Return to the [Summary Table](#).

**Table 5-667. CH7CPREG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG0	R	0h	Multi Chirp 7 Parameters Register 0. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.191 CH7CPREG1 Register (Offset = E0Ch) [Reset = 00000001h]

CH7CPREG1 is shown in [Table 5-668](#).

Return to the [Summary Table](#).

**Table 5-668. CH7CPREG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG1	R	1h	Multi Chirp 7 Parameters Register 1. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.192 CH7CPREG2 Register (Offset = E10h) [Reset = 00000002h]

CH7CPREG2 is shown in [Table 5-669](#).

Return to the [Summary Table](#).

**Table 5-669. CH7CPREG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG2	R	2h	Multi Chirp 7 Parameters Register 2. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.193 CH7CPREG3 Register (Offset = E14h) [Reset = 00000003h]

CH7CPREG3 is shown in [Table 5-670](#).

Return to the [Summary Table](#).

**Table 5-670. CH7CPREG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG3	R	3h	Multi Chirp 7 Parameters Register 3. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.194 CH7CPREG4 Register (Offset = E18h) [Reset = 00000000h]

CH7CPREG4 is shown in [Table 5-671](#).

Return to the [Summary Table](#).

**Table 5-671. CH7CPREG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG4	R	0h	Multi Chirp 7 Parameters Register 4. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.195 CH7CPREG5 Register (Offset = E1Ch) [Reset = 00000000h]

CH7CPREG5 is shown in [Table 5-672](#).

Return to the [Summary Table](#).

**Table 5-672. CH7CPREG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG5	R	0h	Multi Chirp 7 Parameters Register 5. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.196 CH7CPREG6 Register (Offset = E20h) [Reset = 00000000h]

CH7CPREG6 is shown in [Table 5-673](#).

Return to the [Summary Table](#).

**Table 5-673. CH7CPREG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG6	R	0h	Multi Chirp 7 Parameters Register 6. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.197 CH7CPREG7 Register (Offset = E24h) [Reset = 00000000h]

CH7CPREG7 is shown in [Table 5-674](#).

Return to the [Summary Table](#).

**Table 5-674. CH7CPREG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG7	R	0h	Multi Chirp 7 Parameters Register 7. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.198 CH7CPREG8 Register (Offset = E28h) [Reset = 00000000h]

CH7CPREG8 is shown in [Table 5-675](#).

Return to the [Summary Table](#).

**Table 5-675. CH7CPREG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG8	R	0h	Multi Chirp 7 Parameters Register 8. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.199 CH7CPREG9 Register (Offset = E2Ch) [Reset = 00000000h]

CH7CPREG9 is shown in [Table 5-676](#).

Return to the [Summary Table](#).

**Table 5-676. CH7CPREG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG9	R	0h	Multi Chirp 7 Parameters Register 9. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.200 CH7CPREG10 Register (Offset = E30h) [Reset = 00000000h]

CH7CPREG10 is shown in [Table 5-677](#).

Return to the [Summary Table](#).

**Table 5-677. CH7CPREG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG10	R	0h	Multi Chirp 7 Parameters Register 10. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.201 CH7CPREG11 Register (Offset = E34h) [Reset = 00000000h]

CH7CPREG11 is shown in [Table 5-678](#).

Return to the [Summary Table](#).

**Table 5-678. CH7CPREG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG11	R	0h	Multi Chirp 7 Parameters Register 11. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.202 CH7CPREG12 Register (Offset = E38h) [Reset = 00000000h]

CH7CPREG12 is shown in [Table 5-679](#).

Return to the [Summary Table](#).

**Table 5-679. CH7CPREG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG12	R	0h	Multi Chirp 7 Parameters Register 12. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.203 CH7CPREG13 Register (Offset = E3Ch) [Reset = 00000000h]

CH7CPREG13 is shown in [Table 5-680](#).

Return to the [Summary Table](#).

**Table 5-680. CH7CPREG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG13	R	0h	Multi Chirp 7 Parameters Register 13. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.204 CH7CPREG14 Register (Offset = E40h) [Reset = 00000000h]

CH7CPREG14 is shown in [Table 5-681](#).

Return to the [Summary Table](#).

**Table 5-681. CH7CPREG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG14	R	0h	Multi Chirp 7 Parameters Register 14. Refer to Chirp Parameter section for more details (DSS_CP)

### 5.2.6.205 CH7CPREG15 Register (Offset = E44h) [Reset = 00000000h]

CH7CPREG15 is shown in [Table 5-682](#).

Return to the [Summary Table](#).

**Table 5-682. CH7CPREG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CH7CPREG15	R	0h	Multi Chirp 7 Parameters Register 15. Refer to Chirp Parameter section for more details (DSS_CP)



### 5.2.6.206 CH01\_HIL\_CP\_OVERRIDE Register (Offset = E48h) [Reset = 00000000h]

CH01\_HIL\_CP\_OVERRIDE is shown in [Table 5-683](#).

Return to the [Summary Table](#).

**Table 5-683. CH01\_HIL\_CP\_OVERRIDE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	chirp1	R/W	0h	Override data used for Chirp1. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index
15-0	chirp0	R/W	0h	Override data used for Chirp0. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index

### 5.2.6.207 CH23\_HIL\_CP\_OVERRIDE Register (Offset = E4Ch) [Reset = 00000000h]

CH23\_HIL\_CP\_OVERRIDE is shown in [Table 5-684](#).

Return to the [Summary Table](#).

**Table 5-684. CH23\_HIL\_CP\_OVERRIDE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	chirp3	R/W	0h	Override data used for Chirp3. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index
15-0	chirp2	R/W	0h	Override data used for Chirp2. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index

### 5.2.6.208 CH45\_HIL\_CP\_OVERRIDE Register (Offset = E50h) [Reset = 00000000h]

CH45\_HIL\_CP\_OVERRIDE is shown in [Table 5-685](#).

Return to the [Summary Table](#).

**Table 5-685. CH45\_HIL\_CP\_OVERRIDE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	chirp5	R/W	0h	Override data used for Chirp5. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index
15-0	chirp4	R/W	0h	Override data used for Chirp4. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index

### 5.2.6.209 CH67\_HIL\_CP\_OVERRIDE Register (Offset = E54h) [Reset = 00000000h]

CH67\_HIL\_CP\_OVERRIDE is shown in [Table 5-686](#).

Return to the [Summary Table](#).

**Table 5-686. CH67\_HIL\_CP\_OVERRIDE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	chirp7	R/W	0h	Override data used for Chirp7. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index
15-0	chirp6	R/W	0h	Override data used for Chirp6. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index

### 5.2.6.210 CH\_HIL\_CP\_OVERRIDE Register (Offset = E58h) [Reset = 00000000h]

CH\_HIL\_CP\_OVERRIDE is shown in [Table 5-687](#).

Return to the [Summary Table](#).

**Table 5-687. CH\_HIL\_CP\_OVERRIDE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	chirp	R/W	0h	Override data used for Chirp. data[11:0] is used for overriding chirp number data[15:12] is used for overriding chirp profile index

### 5.2.6.211 RSS\_BOOKKEEPING\_CTRL Register (Offset = E5Ch) [Reset = 000XXXXXh]

RSS\_BOOKKEEPING\_CTRL is shown in [Table 5-688](#).

Return to the [Summary Table](#).

**Table 5-688. RSS\_BOOKKEEPING\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-21	RESERVED	R/W	0h	
20	chrp_cnt_rst_src	R/W	0h	0x0: RSS_BOOKKEEPING_CHRP_CNT is reset on every FRAME_START 0x1: RSS_BOOKKEEPING_CHRP_CNT is reset by software
19-17	RESERVED	R/W	0h	
16	chrp_cnt_trig_src	R/W	0h	0x0: RSS_BOOKKEEPING_CHRP_CNT is incremented on every CHIRP_START 0x1: RSS_BOOKKEEPING_CHRP_CNT is incremented on every CHIRP_END
15-13	RESERVED	R/W	0h	
12	frm_cnt_trig_src	R/W	0h	0x0: RSS_BOOKKEEPING_FRM_CNT is incremented on every FRAME_START 0x1: RSS_BOOKKEEPING_FRM_CNT is incremented on every FRAME_END
11-9	RESERVED	R/W	0h	
8	chrp_cnt_rst	R/W	0h	Writing 1'b1 resets the RSS_BOOKKEEPING_CHRP_CNT
7-5	RESERVED	R/W	0h	
4	frm_cnt_rst	R/W	0h	Writing 1'b1 resets the RSS_BOOKKEEPING_FRM_CNT
3-1	RESERVED	R/W	0h	
0	seq_num_rst	R/W	0h	Writing 1'b1 resets the RSS_BOOKKEEPING_SEQ_NUM

### 5.2.6.212 RSS\_APP\_GP Register (Offset = E60h) [Reset = 00000000h]

RSS\_APP\_GP is shown in [Table 5-689](#).

Return to the [Summary Table](#).

**Table 5-689. RSS\_APP\_GP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	reg	R/W	0h	General purpose register for application

### 5.2.6.213 RSS\_BOOKKEEPING\_SEQ\_NUM Register (Offset = E64h) [Reset = 00000000h]

RSS\_BOOKKEEPING\_SEQ\_NUM is shown in [Table 5-690](#).

Return to the [Summary Table](#).

**Table 5-690. RSS\_BOOKKEEPING\_SEQ\_NUM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	reg	R	0h	SEQ_NUM counter is incremented on every read to this register



### 5.2.6.214 RSS\_BOOKKEEPING\_FRM\_CNT Register (Offset = E68h) [Reset = 00000000h]

RSS\_BOOKKEEPING\_FRM\_CNT is shown in [Table 5-691](#).

Return to the [Summary Table](#).

**Table 5-691. RSS\_BOOKKEEPING\_FRM\_CNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	reg	R	0h	FRM_CNT counter is incremented on every frame count trigger source selected using RSS_BOOKKEEPING_CTRL

### 5.2.6.215 RSS\_BOOKKEEPING\_CHRP\_CNT Register (Offset = E6Ch) [Reset = 00000000h]

RSS\_BOOKKEEPING\_CHRP\_CNT is shown in [Table 5-692](#).

Return to the [Summary Table](#).

**Table 5-692. RSS\_BOOKKEEPING\_CHRP\_CNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	reg	R	0h	CHRP_CNT counter is incremented on every chirp count trigger source selected using RSS_BOOKKEEPING_CTRL

### 5.2.6.216 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-693](#).

Return to the [Summary Table](#).

**Table 5-693. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.6.217 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-694](#).

Return to the [Summary Table](#).

**Table 5-694. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.6.218 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-695](#).

Return to the [Summary Table](#).

**Table 5-695. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D

### 5.2.6.219 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-696](#).

Return to the [Summary Table](#).

**Table 5-696. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.6.220 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-697](#).

Return to the [Summary Table](#).

**Table 5-697. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.6.221 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-698](#).

Return to the [Summary Table](#).

**Table 5-698. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D



### 5.2.6.222 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-699](#).

Return to the [Summary Table](#).

**Table 5-699. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

### 5.2.6.223 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-700](#).

Return to the [Summary Table](#).

**Table 5-700. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.6.224 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-701](#).

Return to the [Summary Table](#).

**Table 5-701. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

### 5.2.6.225 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-702](#).

Return to the [Summary Table](#).

**Table 5-702. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.6.226 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-703](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-703. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

### 5.2.6.227 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-704](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-704. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.6.228 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-705](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-705. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.6.229 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-706](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-706. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.



### 5.2.6.230 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-707](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-707. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.6.231 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-708](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-708. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.6.232 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-709](#).

Return to the [Summary Table](#).

EOI register

**Table 5-709. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.6.233 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-710](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-710. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.6.234 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-711](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-711. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault

### 5.2.6.235 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-712](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-712. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.6.236 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-713](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-713. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

## 5.2.7 MSS\_CTRL Registers

Table 5-714 lists the memory-mapped registers for the MSS\_CTRL registers. All register offset addresses not listed in Table 5-714 should be considered as reserved locations and the register contents should not be modified.

**Table 5-714. MSS\_CTRL Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.7.1</a>
4h	MSS_SW_INT		<a href="#">Section 5.2.7.2</a>
8h	MSS_CAPEVNT_SEL		<a href="#">Section 5.2.7.3</a>
Ch	MSS_DMA_REQ_SEL		<a href="#">Section 5.2.7.4</a>
10h	MSS_DMA1_REQ_SEL		<a href="#">Section 5.2.7.5</a>
14h	MSS_IRQ_REQ_SEL		<a href="#">Section 5.2.7.6</a>
18h	MSS_SPI_TRIG_SRC		<a href="#">Section 5.2.7.7</a>
1Ch	MSS_ATCM_MEM_INIT		<a href="#">Section 5.2.7.8</a>
20h	MSS_ATCM_MEM_INIT_DONE		<a href="#">Section 5.2.7.9</a>
24h	MSS_ATCM_MEM_INIT_STATUS		<a href="#">Section 5.2.7.10</a>
28h	MSS_BTCM_MEM_INIT		<a href="#">Section 5.2.7.11</a>
2Ch	MSS_BTCM_MEM_INIT_DONE		<a href="#">Section 5.2.7.12</a>
30h	MSS_BTCM_MEM_INIT_STATUS		<a href="#">Section 5.2.7.13</a>
34h	MSS_L2_MEM_INIT		<a href="#">Section 5.2.7.14</a>
38h	MSS_L2_MEM_INIT_DONE		<a href="#">Section 5.2.7.15</a>
3Ch	MSS_L2_MEM_INIT_STATUS		<a href="#">Section 5.2.7.16</a>
40h	MSS_MAILBOX_MEM_INIT		<a href="#">Section 5.2.7.17</a>
44h	MSS_MAILBOX_MEM_INIT_DONE		<a href="#">Section 5.2.7.18</a>
48h	MSS_MAILBOX_MEM_INIT_STATUS		<a href="#">Section 5.2.7.19</a>
4Ch	MSS_RETRAM_MEM_INIT		<a href="#">Section 5.2.7.20</a>
50h	MSS_RETRAM_MEM_INIT_DONE		<a href="#">Section 5.2.7.21</a>
54h	MSS_RETRAM_MEM_INIT_STATUS		<a href="#">Section 5.2.7.22</a>
58h	MSS_SPIA_MEM_INIT		<a href="#">Section 5.2.7.23</a>
5Ch	MSS_SPIA_MEM_INIT_DONE		<a href="#">Section 5.2.7.24</a>
60h	MSS_SPIA_MEM_INIT_STATUS		<a href="#">Section 5.2.7.25</a>
64h	MSS_SPIB_MEM_INIT		<a href="#">Section 5.2.7.26</a>
68h	MSS_SPIB_MEM_INIT_DONE		<a href="#">Section 5.2.7.27</a>
6Ch	MSS_SPIB_MEM_INIT_STATUS		<a href="#">Section 5.2.7.28</a>
70h	MSS_TPCC_MEMINIT_START		<a href="#">Section 5.2.7.29</a>
74h	MSS_TPCC_MEMINIT_DONE		<a href="#">Section 5.2.7.30</a>
78h	MSS_TPCC_MEMINIT_STATUS		<a href="#">Section 5.2.7.31</a>
7Ch	MSS_GPADC_MEM_INIT		<a href="#">Section 5.2.7.32</a>
80h	MSS_GPADC_MEM_INIT_DONE		<a href="#">Section 5.2.7.33</a>
84h	MSS_GPADC_MEM_INIT_STATUS		<a href="#">Section 5.2.7.34</a>
88h	MSS_SPIA_CFG	RESERVED: Dont Use	
8Ch	MSS_SPIB_CFG		<a href="#">Section 5.2.7.36</a>
90h	MSS_EPWM_CFG		<a href="#">Section 5.2.7.37</a>
94h	MSS_GPIO_CFG		<a href="#">Section 5.2.7.38</a>
9Ch	HW_SPARE_REG1		<a href="#">Section 5.2.7.39</a>
ACh	HW_SPARE_REG2		<a href="#">Section 5.2.7.40</a>



**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
B0h	CCC_ERR_STATUS		<a href="#">Section 5.2.7.41</a>
B4h	CCCA_CFG0		<a href="#">Section 5.2.7.42</a>
B8h	CCCA_CFG1		<a href="#">Section 5.2.7.43</a>
BCh	CCCA_CFG2		<a href="#">Section 5.2.7.44</a>
C0h	CCCA_CFG3		<a href="#">Section 5.2.7.45</a>
C4h	CCCA_CNTVAL		<a href="#">Section 5.2.7.46</a>
C8h	CCCB_CFG0		<a href="#">Section 5.2.7.47</a>
CCh	CCCB_CFG1		<a href="#">Section 5.2.7.48</a>
D0h	CCCB_CFG2		<a href="#">Section 5.2.7.49</a>
D4h	CCCB_CFG3		<a href="#">Section 5.2.7.50</a>
D8h	CCCB_CNTVAL		<a href="#">Section 5.2.7.51</a>
DCh	CCC_DCC_COMMON		<a href="#">Section 5.2.7.52</a>
E0h	R5_GLOBAL_CONFIG		<a href="#">Section 5.2.7.53</a>
E4h	R5_AHB_EN		<a href="#">Section 5.2.7.54</a>
E8h	R5A_AHB_BASE		<a href="#">Section 5.2.7.55</a>
ECh	R5A_AHB_SIZE		<a href="#">Section 5.2.7.56</a>
F0h	R5B_AHB_BASE		<a href="#">Section 5.2.7.57</a>
F4h	R5B_AHB_SIZE		<a href="#">Section 5.2.7.58</a>
F8h	R5_TCM_EXT_ERR_EN		<a href="#">Section 5.2.7.59</a>
FCh	R5_TCM_ERR_EN		<a href="#">Section 5.2.7.60</a>
100h	R5_INIT_TCM		<a href="#">Section 5.2.7.61</a>
104h	R5_TCM_ECC_WRENZ_EN		<a href="#">Section 5.2.7.62</a>
108h	ESM_GATING0		<a href="#">Section 5.2.7.63</a>
10Ch	ESM_GATING1		<a href="#">Section 5.2.7.64</a>
110h	ESM_GATING2		<a href="#">Section 5.2.7.65</a>
114h	ESM_GATING3		<a href="#">Section 5.2.7.66</a>
118h	ESM_GATING4		<a href="#">Section 5.2.7.67</a>
11Ch	ESM_GATING5		<a href="#">Section 5.2.7.68</a>
120h	ESM_GATING6		<a href="#">Section 5.2.7.69</a>
124h	ESM_GATING7		<a href="#">Section 5.2.7.70</a>
128h	ERR_PARITY_ATCM0		<a href="#">Section 5.2.7.71</a>
12Ch	ERR_PARITY_ATCM1		<a href="#">Section 5.2.7.72</a>
130h	ERR_PARITY_B0TCM0		<a href="#">Section 5.2.7.73</a>
134h	ERR_PARITY_B0TCM1		<a href="#">Section 5.2.7.74</a>
138h	ERR_PARITY_B1TCM0		<a href="#">Section 5.2.7.75</a>
13Ch	ERR_PARITY_B1TCM1		<a href="#">Section 5.2.7.76</a>
140h	TCM_PARITY_CTRL		<a href="#">Section 5.2.7.77</a>
144h	TCM_PARITY_ERRFRC		<a href="#">Section 5.2.7.78</a>
148h	HW_SPARE_REG3		<a href="#">Section 5.2.7.79</a>
14Ch	SPIA_IO_CFG		<a href="#">Section 5.2.7.80</a>
150h	SPIB_IO_CFG		<a href="#">Section 5.2.7.81</a>
154h	SPI_HOST_IRQ		<a href="#">Section 5.2.7.82</a>
158h	TPTC_DBS_CONFIG		<a href="#">Section 5.2.7.83</a>
15Ch	TPCC_PARITY_CTRL		<a href="#">Section 5.2.7.84</a>
160h	TPCC_PARITY_STATUS		<a href="#">Section 5.2.7.85</a>

**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
164h	MSS_DBG_ACK_CTL0		<a href="#">Section 5.2.7.86</a>
168h	MSS_DBG_ACK_CTL1		<a href="#">Section 5.2.7.87</a>
16Ch	CPSW_CONTROL		<a href="#">Section 5.2.7.88</a>
170h	MSS_TPCC_A_ERRAGG_MASK		<a href="#">Section 5.2.7.89</a>
174h	MSS_TPCC_A_ERRAGG_STATUS		<a href="#">Section 5.2.7.90</a>
178h	MSS_TPCC_A_ERRAGG_STATUS_RAW		<a href="#">Section 5.2.7.91</a>
17Ch	MSS_TPCC_A_INTAGG_MASK		<a href="#">Section 5.2.7.92</a>
180h	MSS_TPCC_A_INTAGG_STATUS		<a href="#">Section 5.2.7.93</a>
184h	MSS_TPCC_A_INTAGG_STATUS_RAW		<a href="#">Section 5.2.7.94</a>
1A0h	MSS_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.95</a>
1A4h	MSS_CR5A_AXI_RD_BUS_SAFETY_CTL RL		<a href="#">Section 5.2.7.96</a>
1A8h	MSS_CR5A_AXI_RD_BUS_SAFETY_FI		<a href="#">Section 5.2.7.97</a>
1ACh	MSS_CR5A_AXI_RD_BUS_SAFETY_ER R		<a href="#">Section 5.2.7.98</a>
1B0h	MSS_CR5A_AXI_RD_BUS_SAFETY_ER R_STAT_DATA0		<a href="#">Section 5.2.7.99</a>
1B4h	MSS_CR5A_AXI_RD_BUS_SAFETY_ER R_STAT_CMD		<a href="#">Section 5.2.7.100</a>
1B8h	MSS_CR5A_AXI_RD_BUS_SAFETY_ER R_STAT_READ		<a href="#">Section 5.2.7.101</a>
1D4h	MSS_CR5A_AXI_WR_BUS_SAFETY_CTL RL		<a href="#">Section 5.2.7.102</a>
1D8h	MSS_CR5A_AXI_WR_BUS_SAFETY_FI		<a href="#">Section 5.2.7.103</a>
1DCh	MSS_CR5A_AXI_WR_BUS_SAFETY_ER R		<a href="#">Section 5.2.7.104</a>
1E0h	MSS_CR5A_AXI_WR_BUS_SAFETY_ER R_STAT_DATA0		<a href="#">Section 5.2.7.105</a>
1E4h	MSS_CR5A_AXI_WR_BUS_SAFETY_ER R_STAT_CMD		<a href="#">Section 5.2.7.106</a>
1E8h	MSS_CR5A_AXI_WR_BUS_SAFETY_ER R_STAT_WRITE		<a href="#">Section 5.2.7.107</a>
1ECh	MSS_CR5A_AXI_WR_BUS_SAFETY_ER R_STAT_WRITERESP		<a href="#">Section 5.2.7.108</a>
20Ch	MSS_CR5A_AXI_S_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.109</a>
210h	MSS_CR5A_AXI_S_BUS_SAFETY_FI		<a href="#">Section 5.2.7.110</a>
214h	MSS_CR5A_AXI_S_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.111</a>
218h	MSS_CR5A_AXI_S_BUS_SAFETY_ERR_ STAT_DATA0		<a href="#">Section 5.2.7.112</a>
21Ch	MSS_CR5A_AXI_S_BUS_SAFETY_ERR_ STAT_CMD		<a href="#">Section 5.2.7.113</a>
220h	MSS_CR5A_AXI_S_BUS_SAFETY_ERR_ STAT_WRITE		<a href="#">Section 5.2.7.114</a>
224h	MSS_CR5A_AXI_S_BUS_SAFETY_ERR_ STAT_READ		<a href="#">Section 5.2.7.115</a>
228h	MSS_CR5A_AXI_S_BUS_SAFETY_ERR_ STAT_WRITERESP		<a href="#">Section 5.2.7.116</a>
470h	MSS_L2_A_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.117</a>
474h	MSS_L2_A_BUS_SAFETY_FI		<a href="#">Section 5.2.7.118</a>
478h	MSS_L2_A_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.119</a>

**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
47Ch	MSS_L2_A_BUS_SAFETY_ERR_STAT_D ATA0		<a href="#">Section 5.2.7.120</a>
480h	MSS_L2_A_BUS_SAFETY_ERR_STAT_C MD		<a href="#">Section 5.2.7.121</a>
484h	MSS_L2_A_BUS_SAFETY_ERR_STAT_ WRITE		<a href="#">Section 5.2.7.122</a>
488h	MSS_L2_A_BUS_SAFETY_ERR_STAT_R EAD		<a href="#">Section 5.2.7.123</a>
48Ch	MSS_L2_A_BUS_SAFETY_ERR_STAT_ WRITERESP		<a href="#">Section 5.2.7.124</a>
490h	MSS_L2_B_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.125</a>
494h	MSS_L2_B_BUS_SAFETY_FI		<a href="#">Section 5.2.7.126</a>
498h	MSS_L2_B_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.127</a>
49Ch	MSS_L2_B_BUS_SAFETY_ERR_STAT_D ATA0		<a href="#">Section 5.2.7.128</a>
4A0h	MSS_L2_B_BUS_SAFETY_ERR_STAT_C MD		<a href="#">Section 5.2.7.129</a>
4A4h	MSS_L2_B_BUS_SAFETY_ERR_STAT_ WRITE		<a href="#">Section 5.2.7.130</a>
4A8h	MSS_L2_B_BUS_SAFETY_ERR_STAT_R EAD		<a href="#">Section 5.2.7.131</a>
4ACh	MSS_L2_B_BUS_SAFETY_ERR_STAT_ WRITERESP		<a href="#">Section 5.2.7.132</a>
510h	MSS_BUS_SAFETY_SEC_ERR_STAT0		<a href="#">Section 5.2.7.133</a>
514h	MSS_BUS_SAFETY_SEC_ERR_STAT1		<a href="#">Section 5.2.7.134</a>
518h	HW_REG0		<a href="#">Section 5.2.7.135</a>
51Ch	HW_REG1		<a href="#">Section 5.2.7.136</a>
520h	PREVIOUS_NAME		<a href="#">Section 5.2.7.137</a>
524h	HW_REG3		<a href="#">Section 5.2.7.138</a>
528h	HW_REG4		<a href="#">Section 5.2.7.139</a>
52Ch	HW_REG5		<a href="#">Section 5.2.7.140</a>
530h	HW_REG6		<a href="#">Section 5.2.7.141</a>
534h	HW_REG7		<a href="#">Section 5.2.7.142</a>
5B8h	MSS_CR5A_AHB_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.143</a>
5BCh	MSS_CR5A_AHB_BUS_SAFETY_FI		<a href="#">Section 5.2.7.144</a>
5C0h	MSS_CR5A_AHB_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.145</a>
5C4h	MSS_CR5A_AHB_BUS_SAFETY_ERR_S TAT_DATA0		<a href="#">Section 5.2.7.146</a>
5C8h	MSS_CR5A_AHB_BUS_SAFETY_ERR_S TAT_CMD		<a href="#">Section 5.2.7.147</a>
5CCh	MSS_CR5A_AHB_BUS_SAFETY_ERR_S TAT_WRITE		<a href="#">Section 5.2.7.148</a>
5D0h	MSS_CR5A_AHB_BUS_SAFETY_ERR_S TAT_READ		<a href="#">Section 5.2.7.149</a>
5D4h	MSS_CR5A_AHB_BUS_SAFETY_ERR_S TAT_WRITERESP		<a href="#">Section 5.2.7.150</a>
5F8h	DMM_CTRL_REG		<a href="#">Section 5.2.7.151</a>
5FCh	MSS_CR5A_MBOX_WRITE_DONE		<a href="#">Section 5.2.7.152</a>
600h	MSS_CR5A_MBOX_READ_REQ		<a href="#">Section 5.2.7.153</a>
604h	MSS_CR5A_MBOX_READ_DONE		<a href="#">Section 5.2.7.154</a>

**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
614h	MSS_PBIST_KEY_RST		<a href="#">Section 5.2.7.155</a>
618h	MSS_PBIST_REG0		<a href="#">Section 5.2.7.156</a>
61Ch	MSS_PBIST_REG1		<a href="#">Section 5.2.7.157</a>
620h	MSS_PBIST_REG2		<a href="#">Section 5.2.7.158</a>
624h	MSS_QSPI_CONFIG		<a href="#">Section 5.2.7.159</a>
628h	MSS_STC_CONTROL		<a href="#">Section 5.2.7.160</a>
62Ch	MSS_CTI_TRIG_SEL		<a href="#">Section 5.2.7.161</a>
630h	MSS_DBGSS_CTI_TRIG_SEL		<a href="#">Section 5.2.7.162</a>
634h	MSS_BOOT_INFO_REG0		<a href="#">Section 5.2.7.163</a>
638h	MSS_BOOT_INFO_REG1		<a href="#">Section 5.2.7.164</a>
63Ch	MSS_BOOT_INFO_REG2		<a href="#">Section 5.2.7.165</a>
640h	MSS_BOOT_INFO_REG3		<a href="#">Section 5.2.7.166</a>
644h	MSS_BOOT_INFO_REG4		<a href="#">Section 5.2.7.167</a>
648h	MSS_BOOT_INFO_REG5		<a href="#">Section 5.2.7.168</a>
64Ch	MSS_BOOT_INFO_REG6		<a href="#">Section 5.2.7.169</a>
650h	MSS_BOOT_INFO_REG7		<a href="#">Section 5.2.7.170</a>
654h	MSS_TPTC_ECCAGGR_CLK_CNTRL		<a href="#">Section 5.2.7.171</a>
658h	MSS_PERIPH_ERRAGG_MASK0		<a href="#">Section 5.2.7.172</a>
65Ch	MSS_PERIPH_ERRAGG_STATUS0		<a href="#">Section 5.2.7.173</a>
660h	MSS_PERIPH_ERRAGG_STATUS_RAW0		<a href="#">Section 5.2.7.174</a>
664h	MSS_PERIPH_ERRAGG_MASK1		<a href="#">Section 5.2.7.175</a>
668h	MSS_PERIPH_ERRAGG_STATUS1		<a href="#">Section 5.2.7.176</a>
66Ch	MSS_PERIPH_ERRAGG_STATUS_RAW1		<a href="#">Section 5.2.7.177</a>
670h	MSS_DMM_EVENT0_REG		<a href="#">Section 5.2.7.178</a>
674h	MSS_DMM_EVENT1_REG		<a href="#">Section 5.2.7.179</a>
678h	MSS_DMM_EVENT2_REG		<a href="#">Section 5.2.7.180</a>
67Ch	MSS_DMM_EVENT3_REG		<a href="#">Section 5.2.7.181</a>
680h	MSS_DMM_EVENT4_REG		<a href="#">Section 5.2.7.182</a>
684h	MSS_DMM_EVENT5_REG		<a href="#">Section 5.2.7.183</a>
688h	MSS_DMM_EVENT6_REG		<a href="#">Section 5.2.7.184</a>
68Ch	MSS_DMM_EVENT7_REG		<a href="#">Section 5.2.7.185</a>
690h	MSS_DMM_EVENT8_REG		<a href="#">Section 5.2.7.186</a>
694h	MSS_DMM_EVENT9_REG		<a href="#">Section 5.2.7.187</a>
698h	MSS_DMM_EVENT10_REG		<a href="#">Section 5.2.7.188</a>
69Ch	MSS_DMM_EVENT11_REG		<a href="#">Section 5.2.7.189</a>
6A0h	MSS_DMM_EVENT12_REG		<a href="#">Section 5.2.7.190</a>
6A4h	MSS_DMM_EVENT13_REG		<a href="#">Section 5.2.7.191</a>
6A8h	MSS_DMM_EVENT14_REG		<a href="#">Section 5.2.7.192</a>
6ACh	MSS_DMM_EVENT15_REG		<a href="#">Section 5.2.7.193</a>
6B0h	MSS_TPTC_BOUNDARY_CFG		<a href="#">Section 5.2.7.194</a>
6B4h	MSS_TPTC_XID_REORDER_CFG		<a href="#">Section 5.2.7.195</a>
6B8h	GPADC_CTRL		<a href="#">Section 5.2.7.196</a>
6BCh	HW_Sync_FE_CTRL		<a href="#">Section 5.2.7.197</a>
6C0h	DEBUGSS_CSETB_FLUSH		<a href="#">Section 5.2.7.198</a>

**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
6C4h	ANALOG_WU_STATUS_REG_POLARITY_INV		<a href="#">Section 5.2.7.199</a>
6C8h	ANALOG_CLK_STATUS_REG_POLARITY_INV		<a href="#">Section 5.2.7.200</a>
6CCh	ANALOG_WU_STATUS_REG_GRP1_MASK		<a href="#">Section 5.2.7.201</a>
6D0h	ANALOG_CLK_STATUS_REG_GRP1_MASK		<a href="#">Section 5.2.7.202</a>
6D4h	ANALOG_WU_STATUS_REG_GRP2_MASK		<a href="#">Section 5.2.7.203</a>
6D8h	ANALOG_CLK_STATUS_REG_GRP2_MASK		<a href="#">Section 5.2.7.204</a>
6DCh	NERROR_MASK		<a href="#">Section 5.2.7.205</a>
6E0h	MSS2R5SS_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.206</a>
6E4h	MSS2R5SS_BUS_SAFETY_FI		<a href="#">Section 5.2.7.207</a>
6E8h	MSS2R5SS_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.208</a>
6ECh	MSS2R5SS_BUS_SAFETY_ERR_STAT_DATA0		<a href="#">Section 5.2.7.209</a>
6F0h	MSS2R5SS_BUS_SAFETY_ERR_STAT_CMD		<a href="#">Section 5.2.7.210</a>
6F4h	MSS2R5SS_BUS_SAFETY_ERR_STAT_WRITE		<a href="#">Section 5.2.7.211</a>
6F8h	MSS2R5SS_BUS_SAFETY_ERR_STAT_READ		<a href="#">Section 5.2.7.212</a>
6FCh	MSS2R5SS_BUS_SAFETY_ERR_STAT_WRITERESP		<a href="#">Section 5.2.7.213</a>
700h	DSS2R5SS_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.214</a>
704h	DSS2R5SS_BUS_SAFETY_FI		<a href="#">Section 5.2.7.215</a>
708h	DSS2R5SS_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.216</a>
70Ch	DSS2R5SS_BUS_SAFETY_ERR_STAT_DATA0		<a href="#">Section 5.2.7.217</a>
710h	DSS2R5SS_BUS_SAFETY_ERR_STAT_CMD		<a href="#">Section 5.2.7.218</a>
714h	DSS2R5SS_BUS_SAFETY_ERR_STAT_WRITE		<a href="#">Section 5.2.7.219</a>
718h	DSS2R5SS_BUS_SAFETY_ERR_STAT_READ		<a href="#">Section 5.2.7.220</a>
71Ch	DSS2R5SS_BUS_SAFETY_ERR_STAT_WRITERESP		<a href="#">Section 5.2.7.221</a>
720h	MSS_DMM_ACCESS_MODE		<a href="#">Section 5.2.7.222</a>
724h	CPSW_HW_TRIG_CTRL		<a href="#">Section 5.2.7.223</a>
728h	CPSW_HW_TRIG_VAL		<a href="#">Section 5.2.7.224</a>
72Ch	CPSW_TRIG_CAPTURE_COUNT		<a href="#">Section 5.2.7.225</a>
730h	MSS_L2_C_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.226</a>
734h	MSS_L2_C_BUS_SAFETY_FI		<a href="#">Section 5.2.7.227</a>
738h	MSS_L2_C_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.228</a>
73Ch	MSS_L2_C_BUS_SAFETY_ERR_STAT_DATA0		<a href="#">Section 5.2.7.229</a>
740h	MSS_L2_C_BUS_SAFETY_ERR_STAT_CMD		<a href="#">Section 5.2.7.230</a>

**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
744h	MSS_L2_C_BUS_SAFETY_ERR_STAT_WRITE		<a href="#">Section 5.2.7.231</a>
748h	MSS_L2_C_BUS_SAFETY_ERR_STAT_READ		<a href="#">Section 5.2.7.232</a>
74Ch	MSS_L2_C_BUS_SAFETY_ERR_STAT_WRITERESP		<a href="#">Section 5.2.7.233</a>
750h	R5SS2DSS_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.234</a>
754h	R5SS2DSS_BUS_SAFETY_FI		<a href="#">Section 5.2.7.235</a>
758h	R5SS2DSS_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.236</a>
75Ch	R5SS2DSS_BUS_SAFETY_ERR_STAT_DATA0		<a href="#">Section 5.2.7.237</a>
760h	R5SS2DSS_BUS_SAFETY_ERR_STAT_CMD		<a href="#">Section 5.2.7.238</a>
764h	R5SS2DSS_BUS_SAFETY_ERR_STAT_WRITE		<a href="#">Section 5.2.7.239</a>
768h	R5SS2DSS_BUS_SAFETY_ERR_STAT_READ		<a href="#">Section 5.2.7.240</a>
76Ch	R5SS2DSS_BUS_SAFETY_ERR_STAT_WRITERESP		<a href="#">Section 5.2.7.241</a>
770h	R5SS2MSS_BUS_SAFETY_CTRL		<a href="#">Section 5.2.7.242</a>
774h	R5SS2MSS_BUS_SAFETY_FI		<a href="#">Section 5.2.7.243</a>
778h	R5SS2MSS_BUS_SAFETY_ERR		<a href="#">Section 5.2.7.244</a>
77Ch	R5SS2MSS_BUS_SAFETY_ERR_STAT_DATA0		<a href="#">Section 5.2.7.245</a>
780h	R5SS2MSS_BUS_SAFETY_ERR_STAT_CMD		<a href="#">Section 5.2.7.246</a>
784h	R5SS2MSS_BUS_SAFETY_ERR_STAT_WRITE		<a href="#">Section 5.2.7.247</a>
788h	R5SS2MSS_BUS_SAFETY_ERR_STAT_READ		<a href="#">Section 5.2.7.248</a>
78Ch	R5SS2MSS_BUS_SAFETY_ERR_STAT_WRITERESP		<a href="#">Section 5.2.7.249</a>
790h	NW_PACKET_COUNT		<a href="#">Section 5.2.7.250</a>
794h	NW_PACKET_COUNT_RESET		<a href="#">Section 5.2.7.251</a>
798h	CPSW_CRC_PING_ADDR		<a href="#">Section 5.2.7.252</a>
79Ch	CPSW_CRC_PONG_ADDR		<a href="#">Section 5.2.7.253</a>
800h	R5_CONTROL		<a href="#">Section 5.2.7.254</a>
804h	R5_ROM_ECLIPSE		<a href="#">Section 5.2.7.255</a>
808h	R5_COREA_HALT		<a href="#">Section 5.2.7.256</a>
80Ch	R5_COREB_HALT		<a href="#">Section 5.2.7.257</a>
810h	R5_STATUS_REG		<a href="#">Section 5.2.7.258</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.7.259</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.7.260</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.7.261</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.7.262</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.7.263</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.7.264</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.7.265</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.7.266</a>

**Table 5-714. MSS\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.7.267</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.7.268</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.7.269</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.7.270</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.7.271</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.7.272</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.7.273</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.7.274</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.7.275</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.7.276</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.7.277</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.7.278</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.7.279</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-715](#) shows the codes that are used for access types in this section.

**Table 5-715. MSS\_CTRL Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.7.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-716](#).

Return to the [Summary Table](#).

PID register

**Table 5-716. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	



### 5.2.7.2 MSS\_SW\_INT Register (Offset = 4h) [Reset = 00000000h]

MSS\_SW\_INT is shown in [Table 5-717](#).

Return to the [Summary Table](#).

**Table 5-717. MSS\_SW\_INT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-0	pulse	R/W	0h	Write_pulse bit field: writing 1'b1 to each bit will trigger MSS_SW_INT<0-4> respectively to CR5A/B.

### 5.2.7.3 MSS\_CAPEVNT\_SEL Register (Offset = 8h) [Reset = 00000000h]

MSS\_CAPEVNT\_SEL is shown in [Table 5-718](#).

Return to the [Summary Table](#).

**Table 5-718. MSS\_CAPEVNT\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-8	src1	R/W	0h	Writing a value 'N' will select Nth interrupt from CR5A/B interrupt mapping to trigger CAP-EVENT1 to all MSS_RTIs. Example: writing 8'h0A will select 10th interrupt to trigger CAP-EVENT1 to all MSS_RTIs. (which is MSS_RTIB_INT1)
7-0	src0	R/W	0h	Writing a value 'N' will select Nth interrupt from CR5A/B interrupt mapping to trigger CAP-EVENT0 to all MSS_RTIs. Example: writing 8'h0A will select 10th interrupt to trigger CAP-EVENT0 to all MSS_RTIs. (which is MSS_RTIB_INT1)

#### 5.2.7.4 MSS\_DMA\_REQ\_SEL Register (Offset = Ch) [Reset = 00000000h]

MSS\_DMA\_REQ\_SEL is shown in [Table 5-719](#).

Return to the [Summary Table](#).

**Table 5-719. MSS\_DMA\_REQ\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	select	R/W	0h	Reserved for R&D. Do not touch

### 5.2.7.5 MSS\_DMA1\_REQ\_SEL Register (Offset = 10h) [Reset = 00000000h]

MSS\_DMA1\_REQ\_SEL is shown in [Table 5-720](#).

Return to the [Summary Table](#).

**Table 5-720. MSS\_DMA1\_REQ\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	select	R/W	0h	Reserved for R&D. Do not touch

### 5.2.7.6 MSS\_IRQ\_REQ\_SEL Register (Offset = 14h) [Reset = 00000000h]

MSS\_IRQ\_REQ\_SEL is shown in [Table 5-721](#).

Return to the [Summary Table](#).

**Table 5-721. MSS\_IRQ\_REQ\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	select	R/W	0h	Reserved for R&D. Do not touch

### 5.2.7.7 MSS\_SPI\_TRIG\_SRC Register (Offset = 18h) [Reset = 0000XXXXh]

MSS\_SPI\_TRIG\_SRC is shown in [Table 5-722](#).

Return to the [Summary Table](#).

**Table 5-722. MSS\_SPI\_TRIG\_SRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-16	trig_spib	R/W	0h	Writing 1'b1 to each bit will trigger MSS_SPIB Trigger<0-10> respectively
15-2	RESERVED	R/W	0h	
1-0	trig_spia	R/W	0h	RESERVED:Dont Touch

### 5.2.7.8 MSS\_ATCM\_MEM\_INIT Register (Offset = 1Ch) [Reset = 00000000h]

MSS\_ATCM\_MEM\_INIT is shown in [Table 5-723](#).

Return to the [Summary Table](#).

**Table 5-723. MSS\_ATCM\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem_init	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the ATCM banks of CR5A/B. Value in each row is initialized to 0x0C_0000_0000

### 5.2.7.9 MSS\_ATCM\_MEM\_INIT\_DONE Register (Offset = 20h) [Reset = 00000000h]

MSS\_ATCM\_MEM\_INIT\_DONE is shown in [Table 5-724](#).

Return to the [Summary Table](#).

**Table 5-724. MSS\_ATCM\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem_init_done	R/W	0h	This field will be high once initialization of ATCM banks is finished. Writing '1' would clear the bit.



### 5.2.7.10 MSS\_ATCM\_MEM\_INIT\_STATUS Register (Offset = 24h) [Reset = 00000000h]

MSS\_ATCM\_MEM\_INIT\_STATUS is shown in [Table 5-725](#).

Return to the [Summary Table](#).

**Table 5-725. MSS\_ATCM\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem_status	R	0h	1'b0: No initialization is happening for ATCM banks of CR5A/B 1'b1: Initialization is in progress for ATCM banks of CR5A/B

### 5.2.7.11 MSS\_BTCM\_MEM\_INIT Register (Offset = 28h) [Reset = 00000000h]

MSS\_BTCM\_MEM\_INIT is shown in [Table 5-726](#).

Return to the [Summary Table](#).

**Table 5-726. MSS\_BTCM\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem_init	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the B0/1TCM banks of CR5A/B

### 5.2.7.12 MSS\_BTCM\_MEM\_INIT\_DONE Register (Offset = 2Ch) [Reset = 00000000h]

MSS\_BTCM\_MEM\_INIT\_DONE is shown in [Table 5-727](#).

Return to the [Summary Table](#).

**Table 5-727. MSS\_BTCM\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem_init_done	R/W	0h	This field will be high once initialization of B0/1TCM banks is finished. Writing '1' would clear the bit.

### 5.2.7.13 MSS\_BTCM\_MEM\_INIT\_STATUS Register (Offset = 30h) [Reset = 00000000h]

MSS\_BTCM\_MEM\_INIT\_STATUS is shown in [Table 5-728](#).

Return to the [Summary Table](#).

**Table 5-728. MSS\_BTCM\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem_status	R	0h	1'b0: No initialization is happening for B0/1TCM banks of CR5A/B 1'b1: Initialization is in progress for B0/1TCM banks of CR5A/B

#### 5.2.7.14 MSS\_L2\_MEM\_INIT Register (Offset = 34h) [Reset = 00000000h]

MSS\_L2\_MEM\_INIT is shown in [Table 5-729](#).

Return to the [Summary Table](#).

**Table 5-729. MSS\_L2\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	partition2	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the L2 Bank2. Value in each row is initialized to 0x0
1	partition1	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the L2 Bank1. Value in each row is initialized to 0x0
0	partition0	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the L2 Bank0. Value in each row is initialized to 0x0

### 5.2.7.15 MSS\_L2\_MEM\_INIT\_DONE Register (Offset = 38h) [Reset = 00000000h]

MSS\_L2\_MEM\_INIT\_DONE is shown in [Table 5-730](#).

Return to the [Summary Table](#).

**Table 5-730. MSS\_L2\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	partition2	R/W	0h	This field will be high once initialization of L2 bank2 is finished. Writing '1' would clear the bit
1	partition1	R/W	0h	This field will be high once initialization of L2 bank1 is finished. Writing '1' would clear the bit
0	partition0	R/W	0h	This field will be high once initialization of L2 bank0 is finished. Writing '1' would clear the bit

### 5.2.7.16 MSS\_L2\_MEM\_INIT\_STATUS Register (Offset = 3Ch) [Reset = 00000000h]

MSS\_L2\_MEM\_INIT\_STATUS is shown in [Table 5-731](#).

Return to the [Summary Table](#).

**Table 5-731. MSS\_L2\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	partition2	R	0h	1'b0: No initialization is happening for L2 bank2 1'b1: Initialization is in progress for L2 bank2
1	partition1	R	0h	1'b0: No initialization is happening for L2 bank1 1'b1: Initialization is in progress for L2 bank1
0	partition0	R	0h	1'b0: No initialization is happening for L2 bank0 1'b1: Initialization is in progress for L2 bank0

### 5.2.7.17 MSS\_MAILBOX\_MEM\_INIT Register (Offset = 40h) [Reset = 00000000h]

MSS\_MAILBOX\_MEM\_INIT is shown in [Table 5-732](#).

Return to the [Summary Table](#).

**Table 5-732. MSS\_MAILBOX\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the MSS_MBOX. Value in each row is initialized to 0x0



### 5.2.7.18 MSS\_MAILBOX\_MEM\_INIT\_DONE Register (Offset = 44h) [Reset = 00000000h]

MSS\_MAILBOX\_MEM\_INIT\_DONE is shown in [Table 5-733](#).

Return to the [Summary Table](#).

**Table 5-733. MSS\_MAILBOX\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	This field will be high once initialization of MSS_MBOX is finished. Writing '1' would clear the bit

### 5.2.7.19 MSS\_MAILBOX\_MEM\_INIT\_STATUS Register (Offset = 48h) [Reset = 00000000h]

MSS\_MAILBOX\_MEM\_INIT\_STATUS is shown in [Table 5-734](#).

Return to the [Summary Table](#).

**Table 5-734. MSS\_MAILBOX\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	1'b0: No initialization is happening for MSS_MBOX 1'b1: Initialization is in progress for MSS_MBOX

### 5.2.7.20 MSS\_RETRAM\_MEM\_INIT Register (Offset = 4Ch) [Reset = 00000000h]

MSS\_RETRAM\_MEM\_INIT is shown in [Table 5-735](#).

Return to the [Summary Table](#).

**Table 5-735. MSS\_RETRAM\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the MSS_RETRAM. Value in each row is initialized to 0x0

### 5.2.7.21 MSS\_RETRAM\_MEM\_INIT\_DONE Register (Offset = 50h) [Reset = 00000000h]

MSS\_RETRAM\_MEM\_INIT\_DONE is shown in [Table 5-736](#).

Return to the [Summary Table](#).

**Table 5-736. MSS\_RETRAM\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	This field will be high once initialization of MSS_RETRAM is finished. Writing '1' would clear the bit

### 5.2.7.22 MSS\_RETRAM\_MEM\_INIT\_STATUS Register (Offset = 54h) [Reset = 00000000h]

MSS\_RETRAM\_MEM\_INIT\_STATUS is shown in [Table 5-737](#).

Return to the [Summary Table](#).

**Table 5-737. MSS\_RETRAM\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	1'b0: No initialization is happening for MSS_RETRAM 1'b1: Initialization is in progress for MSS_RETRAM

### 5.2.7.23 MSS\_SPIA\_MEM\_INIT Register (Offset = 58h) [Reset = 00000000h]

MSS\_SPIA\_MEM\_INIT is shown in [Table 5-738](#).

Return to the [Summary Table](#).

**Table 5-738. MSS\_SPIA\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	RESERVED:Dont Touch

#### 5.2.7.24 MSS\_SPIA\_MEM\_INIT\_DONE Register (Offset = 5Ch) [Reset = 00000000h]

MSS\_SPIA\_MEM\_INIT\_DONE is shown in [Table 5-739](#).

Return to the [Summary Table](#).

**Table 5-739. MSS\_SPIA\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	RESERVED: Dont Use

### 5.2.7.25 MSS\_SPIA\_MEM\_INIT\_STATUS Register (Offset = 60h) [Reset = 00000000h]

MSS\_SPIA\_MEM\_INIT\_STATUS is shown in [Table 5-740](#).

Return to the [Summary Table](#).

**Table 5-740. MSS\_SPIA\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	RESERVED: Dont Use



### 5.2.7.26 MSS\_SPIB\_MEM\_INIT Register (Offset = 64h) [Reset = 00000000h]

MSS\_SPIB\_MEM\_INIT is shown in [Table 5-741](#).

Return to the [Summary Table](#).

**Table 5-741. MSS\_SPIB\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the MSS_SPIB. Value in each row is initialized to 0x0

### 5.2.7.27 MSS\_SPIB\_MEM\_INIT\_DONE Register (Offset = 68h) [Reset = 00000000h]

MSS\_SPIB\_MEM\_INIT\_DONE is shown in [Table 5-742](#).

Return to the [Summary Table](#).

**Table 5-742. MSS\_SPIB\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	This field will be high once initialization of MSS_SPIB is finished. Writing '1' would clear the bit

### 5.2.7.28 MSS\_SPIB\_MEM\_INIT\_STATUS Register (Offset = 6Ch) [Reset = 00000000h]

MSS\_SPIB\_MEM\_INIT\_STATUS is shown in [Table 5-743](#).

Return to the [Summary Table](#).

**Table 5-743. MSS\_SPIB\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	1'b0: No initialization is happening for MSS_SPIB 1'b1: Initialization is in progress for MSS_SPIB

### 5.2.7.29 MSS\_TPCC\_MEMINIT\_START Register (Offset = 70h) [Reset = 00000000h]

MSS\_TPCC\_MEMINIT\_START is shown in [Table 5-744](#).

Return to the [Summary Table](#).

**Table 5-744. MSS\_TPCC\_MEMINIT\_START Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	tpcc_a_meminit_start	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the MSS_TPCCA

### 5.2.7.30 MSS\_TPCC\_MEMINIT\_DONE Register (Offset = 74h) [Reset = 00000000h]

MSS\_TPCC\_MEMINIT\_DONE is shown in [Table 5-745](#).

Return to the [Summary Table](#).

**Table 5-745. MSS\_TPCC\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	tpcc_a_meminit_done	R/W	0h	This field will be high once initialization of MSS_TPCCA is finished. Writing '1' would clear the bit

### 5.2.7.31 MSS\_TPCC\_MEMINIT\_STATUS Register (Offset = 78h) [Reset = 00000000h]

MSS\_TPCC\_MEMINIT\_STATUS is shown in [Table 5-746](#).

Return to the [Summary Table](#).

**Table 5-746. MSS\_TPCC\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	tpcc_a_meminit_status	R	0h	1'b0: No initialization is happening for MSS_TPCCA 1'b1: Initialization is in progress for MSS_TPCCB

### 5.2.7.32 MSS\_GPADC\_MEM\_INIT Register (Offset = 7Ch) [Reset = 00000000h]

MSS\_GPADC\_MEM\_INIT is shown in [Table 5-747](#).

Return to the [Summary Table](#).

**Table 5-747. MSS\_GPADC\_MEM\_INIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_init	R/W	0h	Write_pulse bit field: Writing 1'b1 will start initializing the MSS_GPADC_DATA_MEM. Value in each row is initialized to 0x00_0000_03FF

### 5.2.7.33 MSS\_GPADC\_MEM\_INIT\_DONE Register (Offset = 80h) [Reset = 00000000h]

MSS\_GPADC\_MEM\_INIT\_DONE is shown in [Table 5-748](#).

Return to the [Summary Table](#).

**Table 5-748. MSS\_GPADC\_MEM\_INIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	mem0_done	R/W	0h	This field will be high once initialization of MSS_GPADC_DATA_MEM is finished. Writing '1' would clear the bit



### 5.2.7.34 MSS\_GPADC\_MEM\_INIT\_STATUS Register (Offset = 84h) [Reset = 00000000h]

MSS\_GPADC\_MEM\_INIT\_STATUS is shown in [Table 5-749](#).

Return to the [Summary Table](#).

**Table 5-749. MSS\_GPADC\_MEM\_INIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	mem0_status	R	0h	1'b0: No initialization is happening for MSS_GPADC_DATA_MEM 1'b1: Initialization is in progress for MSS_GPADC_DATA_MEM

### 5.2.7.35 MSS\_SPIA\_CFG Register (Offset = 88h) [Reset = 00XXXXXXh]

MSS\_SPIA\_CFG is shown in [Table 5-750](#).

Return to the [Summary Table](#).

RESERVED: Dont Use

**Table 5-750. MSS\_SPIA\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	spia_int_trig_polarity	R/W	0h	RESERVED:Dont Use
23-17	RESERVED	R/W	0h	
16	spia_trig_gate_en	R/W	0h	RESERVED:Dont Use
15-9	RESERVED	R/W	0h	
8	spia_cs_trigsrc_en	R/W	0h	RESERVED:Dont Use
7-3	RESERVED	R/W	0h	
2-0	spiasync2sen	R/W	0h	RESERVED:Dont Use

### 5.2.7.36 MSS\_SPIB\_CFG Register (Offset = 8Ch) [Reset = 00XXXXXXh]

MSS\_SPIB\_CFG is shown in [Table 5-751](#).

Return to the [Summary Table](#).

**Table 5-751. MSS\_SPIB\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	spib_int_trig_polarity	R/W	0h	SPIB trigger source polarity select. 0 - Polarity 0, 1 -Polarity 1
23-17	RESERVED	R/W	0h	
16	spib_trig_gate_en	R/W	0h	When set the TRIGGER s are un-gated only when chip-select is active
15-9	RESERVED	R/W	0h	
8	spib_cs_trigsrc_en	R/W	0h	MIBSPIB CS Trigger SRC enable 1 : Use CS as trigger source
7-3	RESERVED	R/W	0h	
2-0	spibsync2sen	R/W	0h	Donot touch the field. Used as Tie-off for IP-config.

### 5.2.7.37 MSS\_EPWM\_CFG Register (Offset = 90h) [Reset = 0F000000h]

MSS\_EPWM\_CFG is shown in [Table 5-752](#).

Return to the [Summary Table](#).

**Table 5-752. MSS\_EPWM\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	epwm_config	R/W	0F000000h	bit0: SW syncin for EPWM1 bit1: SW syncin for EPWM2 bit2: SW syncin for EPWM3 bit8:9 : select bits for EPWM1 '0' : external syncin '1' : reserved '2' : sw syncin '3' : reserved bit10:11 : select bits for EPWM2 '0' : external syncin '1' : chained from EPWM1 '2' : sw syncin '3' : reserved bit12:13 : select bits for EPWM3 '0' : external syncin '1' : chained from EPWM2 '2' : sw syncin '3' : reserved bit24:TBCLKEN for EPWM1 bit25:TBCLKEN for EPWM2 bit26:TBCLKEN for EPWM3

### 5.2.7.38 MSS\_GPIO\_CFG Register (Offset = 94h) [Reset = 00000000h]

MSS\_GPIO\_CFG is shown in [Table 5-753](#).

Return to the [Summary Table](#).

**Table 5-753. MSS\_GPIO\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	gio_config	R/W	0h	bit0 : writing '1' will select negedge for pulse generation of GIO_PAD_INT0 to IRQ bit1 : writing '1' will select negedge for pulse generation of GIO_PAD_INT1 to IRQ bit2 : writing '1' will select negedge for pulse generation of GIO_PAD_INT2 to IRQ bit3 : writing '1' will select negedge for pulse generation of GIO_PAD_INT3 to IRQ bit4 : writing '1' will select negedge for pulse generation of GIO_PAD_INT4 to IRQ bit5 : writing '1' will select negedge for pulse generation of GIO_PAD_INT5 to IRQ bit6 : writing '1' will select negedge for pulse generation of GIO_PAD_INT6 to IRQ bit7 : writing '1' will select negedge for pulse generation of GIO_PAD_INT7 to IRQ

### 5.2.7.39 HW\_SPARE\_REG1 Register (Offset = 9Ch) [Reset = 00000000h]

HW\_SPARE\_REG1 is shown in [Table 5-754](#).

Return to the [Summary Table](#).

**Table 5-754. HW\_SPARE\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Resereved for R&D

### 5.2.7.40 HW\_SPARE\_REG2 Register (Offset = ACh) [Reset = 00000000h]

HW\_SPARE\_REG2 is shown in [Table 5-755](#).

Return to the [Summary Table](#).

**Table 5-755. HW\_SPARE\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Resereved for R&D

### 5.2.7.41 CCC\_ERR\_STATUS Register (Offset = B0h) [Reset = 0000XX00h]

CCC\_ERR\_STATUS is shown in [Table 5-756](#).

Return to the [Summary Table](#).

**Table 5-756. CCC\_ERR\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	cccb_errot_status	R	0h	CCCB Error Status (for Debug) {3'd0, counter_error, counter_done, timeout_error, counter_error, counter_done}
15-8	RESERVED	R	0h	
7-0	ccca_errot_status	R	0h	CCCA Error Status (for Debug) {3'd0, counter_error, counter_done, timeout_error, counter_error, counter_done}



### 5.2.7.42 CCCA\_CFG0 Register (Offset = B4h) [Reset = 00000000h]

CCCA\_CFG0 is shown in [Table 5-757](#).

Return to the [Summary Table](#).

**Table 5-757. CCCA\_CFG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ccca_margin_count	R/W	0h	Margin value for clock comparison in terms of counter1 clock. CCC error will not be generated if counter1 counter value is within count1_expected_val +/- MARGIN_COUNT
15-9	RESERVED	R/W	0h	Not used
8	ccca_single_shot_mode	R/W	0h	1: Single shot mode 0: Continuous mode
7	ccca_enable_module	R/W	0h	1'b1: Enables CCCA 1'b0: Disables CCCA
6	ccca_disable_clocks	R/W	0h	1: Clock gated to counter0 and counter1 0: Normal mode
5-3	ccca_clk1_sel	R/W	0h	Selection for Clock 1 0: Select clock0_src0 as source for counter1 1: Select clock0_src1 as source for counter1 2: Select clock0_src2 as source for counter1 ... 7: Select clock0_src7 as source for counter1
2-0	ccca_clk0_sel	R/W	0h	Selection for Clock 0 0: Select clock0_src0 as source for counter0 1: Select clock0_src1 as source for counter0 2: Select clock0_src2 as source for counter0 ... 7: Select clock0_src7 as source for counter0

### 5.2.7.43 CCCA\_CFG1 Register (Offset = B8h) [Reset = 00000000h]

CCCA\_CFG1 is shown in [Table 5-758](#).

Return to the [Summary Table](#).

**Table 5-758. CCCA\_CFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ccca_cfg	R/W	0h	count0_expiry_val Counter 1 is compared for count1_expected_val +/- MARGIN_COUNT when counter0 expires after counting down from count0_expiry_val to 0

#### 5.2.7.44 CCCA\_CFG2 Register (Offset = BCh) [Reset = 00000000h]

CCCA\_CFG2 is shown in [Table 5-759](#).

Return to the [Summary Table](#).

**Table 5-759. CCCA\_CFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ccca_cfg	R/W	0h	count1_expected_val Expected value of counter 1 when counter 0 expires after counting down from count0_expiry value

### 5.2.7.45 CCCA\_CFG3 Register (Offset = C0h) [Reset = 00000000h]

CCCA\_CFG3 is shown in [Table 5-760](#).

Return to the [Summary Table](#).

**Table 5-760. CCCA\_CFG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ccca_cfg	R/W	0h	Timeout Error Counter value in counter1 clock

### 5.2.7.46 CCCA\_CNTVAL Register (Offset = C4h) [Reset = 00000000h]

CCCA\_CNTVAL is shown in [Table 5-761](#).

Return to the [Summary Table](#).

**Table 5-761. CCCA\_CNTVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ccca_cfg	R	0h	count1_val_out Real time value of counter1

### 5.2.7.47 CCCB\_CFG0 Register (Offset = C8h) [Reset = 00000000h]

CCCB\_CFG0 is shown in [Table 5-762](#).

Return to the [Summary Table](#).

**Table 5-762. CCCB\_CFG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	cccb_margin_count	R/W	0h	Margin value for clock comparison in terms of counter1 clock. CCC error will not be generated if counter1 counter value is within count1_expected_val +/- MARGIN_COUNT
15-9	RESERVED	R/W	0h	Not used
8	cccb_single_shot_mode	R/W	0h	1: Single shot mode 0: Continuous mode
7	cccb_enable_module	R/W	0h	1'b1: Enables CCCB 1'b0: Disables CCCB
6	cccb_disable_clocks	R/W	0h	1: Clock gated to counter0 and counter1 0: Normal mode
5-3	CCCB_clk1_sel	R/W	0h	Selection for Clock 1 0: Select clock0_src0 as source for counter1 1: Select clock0_src1 as source for counter1 2: Select clock0_src2 as source for counter1 ... 7: Select clock0_src7 as source for counter1
2-0	CCCB_clk0_sel	R/W	0h	Selection for Clock 0 0: Select clock0_src0 as source for counter0 1: Select clock0_src1 as source for counter0 2: Select clock0_src2 as source for counter0 ... 7: Select clock0_src7 as source for counter0

### 5.2.7.48 CCCB\_CFG1 Register (Offset = CCh) [Reset = 00000000h]

CCCB\_CFG1 is shown in [Table 5-763](#).

Return to the [Summary Table](#).

**Table 5-763. CCCB\_CFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	cccb_cfg	R/W	0h	count0_expiry_val Counter 1 is compared for count1_expected_val +/- MARGIN_COUNT when counter0 expires after counting down from count0_expiry_val to 0

### 5.2.7.49 CCCB\_CFG2 Register (Offset = D0h) [Reset = 00000000h]

CCCB\_CFG2 is shown in [Table 5-764](#).

Return to the [Summary Table](#).

**Table 5-764. CCCB\_CFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	cccb_cfg	R/W	0h	count1_expected_val Expected value of counter 1 when counter 0 expires after counting down from count0_expiry value



### 5.2.7.50 CCCB\_CFG3 Register (Offset = D4h) [Reset = 00000000h]

CCCB\_CFG3 is shown in [Table 5-765](#).

Return to the [Summary Table](#).

**Table 5-765. CCCB\_CFG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	cccb_cfg	R/W	0h	Timeout Error Counter value in counter1 clock

### 5.2.7.51 CCCB\_CNTVAL Register (Offset = D8h) [Reset = 00000000h]

CCCB\_CNTVAL is shown in [Table 5-766](#).

Return to the [Summary Table](#).

**Table 5-766. CCCB\_CNTVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	cccb_cfg	R	0h	count1_val_out Real time value of counter1

### 5.2.7.52 CCC\_DCC\_COMMON Register (Offset = DCh) [Reset = 00000XXXh]

CCC\_DCC\_COMMON is shown in [Table 5-767](#).

Return to the [Summary Table](#).

**Table 5-767. CCC\_DCC\_COMMON Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12	enable_cccb_err_nmi	R/W	0h	1'b0: Enable CCCB error to generate NMI. 1'b1: disables CCCB error to generate NMI.
11-9	RESERVED	R/W	0h	
8	enable_cccb_err_rstn	R/W	0h	1'b0: Enable CCCB error to generate WD restrn. 1'b1: disables CCCB error to generate WD restrn.
7-0	RESERVED	R/W	0h	

### 5.2.7.53 R5\_GLOBAL\_CONFIG Register (Offset = E0h) [Reset = 00000000h]

R5\_GLOBAL\_CONFIG is shown in [Table 5-768](#).

Return to the [Summary Table](#).

**Table 5-768. R5\_GLOBAL\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	teinit	R/W	0h	Exception handling state at reset. 0-ARM 1-Thumb

### 5.2.7.54 R5\_AHB\_EN Register (Offset = E4h) [Reset = 0007XXXXh]

R5\_AHB\_EN is shown in [Table 5-769](#).

Return to the [Summary Table](#).

**Table 5-769. R5\_AHB\_EN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	cpu1_ahb_init	R/W	7h	Ti internal Register. Modifying this register is not recommended Signal decides whehter ahb interface is enabled or not.
15-3	RESERVED	R/W	0h	
2-0	cpu0_ahb_init	R/W	7h	Ti internal Register. Modifying this register is not recommended Signal decides whehter ahb interface is enabled or not.

### 5.2.7.55 R5A\_AHB\_BASE Register (Offset = E8h) [Reset = 00000000h]

R5A\_AHB\_BASE is shown in [Table 5-770](#).

Return to the [Summary Table](#).

**Table 5-770. R5A\_AHB\_BASE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-0	ahb_base	R/W	0h	Ti internal Register. Modifying this register is not recommended Decides the base address of ahb region

### 5.2.7.56 R5A\_AHB\_SIZE Register (Offset = ECh) [Reset = 00000012h]

R5A\_AHB\_SIZE is shown in [Table 5-771](#).

Return to the [Summary Table](#).

**Table 5-771. R5A\_AHB\_SIZE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-0	ahb_size	R/W	12h	Ti internal Register. Modifying this register is not recommended Code for selecting size for ahb. b00011 4KB b00100 8KB b00101 16KB b00110 32KB b00111 64KB b01000 128KB b01001 256KB b01010 512KB b01011 1MB b01100 2MB b01101 4MB b01110 8MB b01111 16MB b10000 32MB b10001 64MB b10010 128MB b10011 256MB b10100 512MB b10101 1GB b10110 2GB b10111 4GB

### 5.2.7.57 R5B\_AHB\_BASE Register (Offset = F0h) [Reset = 00000000h]

R5B\_AHB\_BASE is shown in [Table 5-772](#).

Return to the [Summary Table](#).

**Table 5-772. R5B\_AHB\_BASE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	
19-0	ahb_base	R/W	0h	Ti internal Register. Modifying this register is not recommended Decides the base address of ahb region



### 5.2.7.58 R5B\_AHB\_SIZE Register (Offset = F4h) [Reset = 00000012h]

R5B\_AHB\_SIZE is shown in [Table 5-773](#).

Return to the [Summary Table](#).

**Table 5-773. R5B\_AHB\_SIZE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4-0	ahb_size	R/W	12h	Ti internal Register. Modifying this register is not recommended Code for selecting size for ahb. b00011 4KB b00100 8KB b00101 16KB b00110 32KB b00111 64KB b01000 128KB b01001 256KB b01010 512KB b01011 1MB b01100 2MB b01101 4MB b01110 8MB b01111 16MB b10000 32MB b10001 64MB b10010 128MB b10011 256MB b10100 512MB b10101 1GB b10110 2GB b10111 4GB

### 5.2.7.59 R5\_TCM\_EXT\_ERR\_EN Register (Offset = F8h) [Reset = 0007XXXXh]

R5\_TCM\_EXT\_ERR\_EN is shown in [Table 5-774](#).

Return to the [Summary Table](#).

**Table 5-774. R5\_TCM\_EXT\_ERR\_EN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	cpu1_tcm	R/W	7h	Ti internal Register. Modifying this register is not recommended TCMs external error enable. Tie each bit high to enable the external error signal for each TCM at reset
15-3	RESERVED	R/W	0h	
2-0	cpu0_tcm	R/W	7h	Ti internal Register. Modifying this register is not recommended TCMs external error enable. Tie each bit high to enable the external error signal for each TCM at reset

### 5.2.7.60 R5\_TCM\_ERR\_EN Register (Offset = FCh) [Reset = 0000XXXXh]

R5\_TCM\_ERR\_EN is shown in [Table 5-775](#).

Return to the [Summary Table](#).

**Table 5-775. R5\_TCM\_ERR\_EN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	cpu1_tcm	R/W	0h	Ti internal Register. Modifying this register is not recommended TCMs ECC check enable. Tie each bit high to enable ECC checking on appropriate TCM
15-3	RESERVED	R/W	0h	
2-0	cpu0_tcm	R/W	0h	Ti internal Register. Modifying this register is not recommended TCMs ECC check enable. Tie each bit high to enable ECC checking on appropriate TCM

### 5.2.7.61 R5\_INIT\_TCM Register (Offset = 100h) [Reset = 007XXXXXh]

R5\_INIT\_TCM is shown in [Table 5-776](#).

Return to the [Summary Table](#).

**Table 5-776. R5\_INIT\_TCM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-20	lockzram_cpu1	R/W	7h	Ti internal Register. Modifying this register is not recommended When HIGH ATCM base address at reset is 0x0 when LOW BTCM base address at reset is 0x0
19	RESERVED	R/W	0h	
18-16	tcmb_cpu1	R/W	7h	Ti internal Register. Modifying this register is not recommended When HIGH enables BTCM interface out of reset
15	RESERVED	R/W	0h	
14-12	tcma_cpu1	R/W	7h	Ti internal Register. Modifying this register is not recommended When HIGH enables ATCM interface out of reset
11	RESERVED	R/W	0h	
10-8	lockzram_cpu0	R/W	7h	Ti internal Register. Modifying this register is not recommended When HIGH ATCM base address at reset is 0x0 when LOW BTCM base address at reset is 0x0
7	RESERVED	R/W	0h	
6-4	tcmb_cpu0	R/W	7h	Ti internal Register. Modifying this register is not recommended When HIGH enables BTCM interface out of reset
3	RESERVED	R/W	0h	
2-0	tcma_cpu0	R/W	7h	Ti internal Register. Modifying this register is not recommended When HIGH enables ATCM interface out of reset

### 5.2.7.62 R5\_TCM\_ECC\_WRENZ\_EN Register (Offset = 104h) [Reset = 007XXXXXh]

R5\_TCM\_ECC\_WRENZ\_EN is shown in [Table 5-777](#).

Return to the [Summary Table](#).

**Table 5-777. R5\_TCM\_ECC\_WRENZ\_EN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-20	cpu1_tcmb1_wrenz_en	R/W	7h	RESERVED: Dont Use
19	RESERVED	R/W	0h	
18-16	cpu1_tcmb0_wrenz_en	R/W	7h	RESERVED: Dont Use
15	RESERVED	R/W	0h	
14-12	cpu1_tcma_wrenz_en	R/W	7h	RESERVED: Dont Use
11	RESERVED	R/W	0h	
10-8	cpu0_tcmb1_wrenz_en	R/W	7h	writing '000' blocks the writes to ECC-bits of TCMB0-RAM of CR5A. Writing '111' unblocks the writes to ECC-bits of TCMB1-RAM of CR5A
7	RESERVED	R/W	0h	
6-4	cpu0_tcmb0_wrenz_en	R/W	7h	writing '000' blocks the writes to ECC-bits of TCMB0-RAM of CR5A. Writing '111' unblocks the writes to ECC-bits of TCMB0-RAM of CR5A
3	RESERVED	R/W	0h	
2-0	cpu0_tcma_wrenz_en	R/W	7h	writing '000' blocks the writes to ECC-bits of TCMA-RAM of CR5A. Writing '111' unblocks the writes to ECC-bits of TCMA-RAM of CR5A

### 5.2.7.63 ESM\_GATING0 Register (Offset = 108h) [Reset = FFFFFFFFh]

ESM\_GATING0 is shown in [Table 5-778](#).

Return to the [Summary Table](#).

**Table 5-778. ESM\_GATING0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_0 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_1 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_7

### 5.2.7.64 ESM\_GATING1 Register (Offset = 10Ch) [Reset = FFFFFFFFh]

ESM\_GATING1 is shown in [Table 5-779](#).

Return to the [Summary Table](#).

**Table 5-779. ESM\_GATING1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_8 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_9 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_15

### 5.2.7.65 ESM\_GATING2 Register (Offset = 110h) [Reset = FFFFFFFFh]

ESM\_GATING2 is shown in [Table 5-780](#).

Return to the [Summary Table](#).

**Table 5-780. ESM\_GATING2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_16 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_17 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_23



### 5.2.7.66 ESM\_GATING3 Register (Offset = 114h) [Reset = FFFFFFFFh]

ESM\_GATING3 is shown in [Table 5-781](#).

Return to the [Summary Table](#).

**Table 5-781. ESM\_GATING3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP2_ERROR_24 bit7:4 : writing '000' will ungate the ESM_GRP2_ERROR_25 bit31:28 : writing '000' will ungate the ESM_GRP2_ERROR_31

### 5.2.7.67 ESM\_GATING4 Register (Offset = 118h) [Reset = FFFFFFFFh]

ESM\_GATING4 is shown in [Table 5-782](#).

Return to the [Summary Table](#).

**Table 5-782. ESM\_GATING4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_0 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_1 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_7

### 5.2.7.68 ESM\_GATING5 Register (Offset = 11Ch) [Reset = FFFFFFFFh]

ESM\_GATING5 is shown in [Table 5-783](#).

Return to the [Summary Table](#).

**Table 5-783. ESM\_GATING5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_8 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_9 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_15

### 5.2.7.69 ESM\_GATING6 Register (Offset = 120h) [Reset = FFFFFFFFh]

ESM\_GATING6 is shown in [Table 5-784](#).

Return to the [Summary Table](#).

**Table 5-784. ESM\_GATING6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_16 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_17 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_23

### 5.2.7.70 ESM\_GATING7 Register (Offset = 124h) [Reset = FFFFFFFFh]

ESM\_GATING7 is shown in [Table 5-785](#).

Return to the [Summary Table](#).

**Table 5-785. ESM\_GATING7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	esm_gating	R/W	FFFFFFFh	bit3:0 : writing '000' will ungate the ESM_GRP3_ERROR_24 bit7:4 : writing '000' will ungate the ESM_GRP3_ERROR_25 bit31:28 : writing '000' will ungate the ESM_GRP3_ERROR_31

### 5.2.7.71 ERR\_PARITY\_ATCM0 Register (Offset = 128h) [Reset = 00000000h]

ERR\_PARITY\_ATCM0 is shown in [Table 5-786](#).

Return to the [Summary Table](#).

**Table 5-786. ERR\_PARITY\_ATCM0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	addr	R	0h	Address latched when parity error is occurred for ATCM of CR5A

### 5.2.7.72 ERR\_PARITY\_ATCM1 Register (Offset = 12Ch) [Reset = 00000000h]

ERR\_PARITY\_ATCM1 is shown in [Table 5-787](#).

Return to the [Summary Table](#).

**Table 5-787. ERR\_PARITY\_ATCM1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	addr	R	0h	RESERVED: Dont Use

### 5.2.7.73 ERR\_PARITY\_B0TCM0 Register (Offset = 130h) [Reset = 00000000h]

ERR\_PARITY\_B0TCM0 is shown in [Table 5-788](#).

Return to the [Summary Table](#).

**Table 5-788. ERR\_PARITY\_B0TCM0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	addr	R	0h	Address latched when parity error is occurred for B0TCM of CR5A



#### 5.2.7.74 ERR\_PARITY\_B0TCM1 Register (Offset = 134h) [Reset = 00000000h]

ERR\_PARITY\_B0TCM1 is shown in [Table 5-789](#).

Return to the [Summary Table](#).

**Table 5-789. ERR\_PARITY\_B0TCM1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	addr	R	0h	RESERVED: Dont Use

### 5.2.7.75 ERR\_PARITY\_B1TCM0 Register (Offset = 138h) [Reset = 00000000h]

ERR\_PARITY\_B1TCM0 is shown in [Table 5-790](#).

Return to the [Summary Table](#).

**Table 5-790. ERR\_PARITY\_B1TCM0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	addr	R	0h	Address latched when parity error is occurred for B1TCM of CR5A

### 5.2.7.76 ERR\_PARITY\_B1TCM1 Register (Offset = 13Ch) [Reset = 00000000h]

ERR\_PARITY\_B1TCM1 is shown in [Table 5-791](#).

Return to the [Summary Table](#).

**Table 5-791. ERR\_PARITY\_B1TCM1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	addr	R	0h	RESERVED: Dont Use

### 5.2.7.77 TCM\_PARITY\_CTRL Register (Offset = 140h) [Reset = 000XXXXXh]

TCM\_PARITY\_CTRL is shown in [Table 5-792](#).

Return to the [Summary Table](#).

**Table 5-792. TCM\_PARITY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-20	b1tcm1_erraddr_clr	R/W	0h	RESERVED: Dont Use
19	RESERVED	R/W	0h	
18-16	b1tcm0_erraddr_clr	R/W	0h	Write pulse bit field: writing 3'b111 clears the Address latched after parity error for B1TCM of CR5A
15	RESERVED	R/W	0h	
14-12	b0cm1_erraddr_clr	R/W	0h	RESERVED: Dont Use
11	RESERVED	R/W	0h	
10-8	b0tcm0_erraddr_clr	R/W	0h	RESERVED: Dont Use
7	RESERVED	R/W	0h	
6-4	atcm1_erraddr_clr	R/W	0h	RESERVED: Dont Use
3	RESERVED	R/W	0h	
2-0	atcm0_erraddr_clr	R/W	0h	Pulse bit-field writing 3'b111 clears the Address latched after parity error for ATCM of CR5A

### 5.2.7.78 TCM\_PARITY\_ERRFRC Register (Offset = 144h) [Reset = 000XXXXh]

TCM\_PARITY\_ERRFRC is shown in [Table 5-793](#).

Return to the [Summary Table](#).

**Table 5-793. TCM\_PARITY\_ERRFRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	0h	
22-20	b1tcm1	R/W	0h	RESERVED: Dont Use
19	RESERVED	R/W	0h	
18-16	b1tcm0	R/W	0h	Write pulse bit field: writing 3'b111 forces a parity error for B1TCM of CR5A
15	RESERVED	R/W	0h	
14-12	b0tcm1	R/W	0h	RESERVED: Dont Use
11	RESERVED	R/W	0h	
10-8	b0tcm0	R/W	0h	Write pulse bit field: writing 3'b111 forces a parity error for B0TCM of CR5A
7	RESERVED	R/W	0h	
6-4	atcm1	R/W	0h	RESERVED: Dont Use
3	RESERVED	R/W	0h	
2-0	atcm0	R/W	0h	Write pulse bit field: writing 3'b111 forces a parity error for ATCM of CR5A

### 5.2.7.79 HW\_SPARE\_REG3 Register (Offset = 148h) [Reset = 00000000h]

HW\_SPARE\_REG3 is shown in [Table 5-794](#).

Return to the [Summary Table](#).

**Table 5-794. HW\_SPARE\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Resereved for R&D

### 5.2.7.80 SPIA\_IO\_CFG Register (Offset = 14Ch) [Reset = 0000XXXXh]

SPIA\_IO\_CFG is shown in [Table 5-795](#).

Return to the [Summary Table](#).

**Table 5-795. SPIA\_IO\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	miso_oen_by_cs	R/W	0h	RESERVED:Dont Use
15-11	RESERVED	R/W	0h	
10-8	cs_pol	R/W	0h	RESERVED:Dont Use
7-3	RESERVED	R/W	0h	
2-0	cs_deact	R/W	0h	RESERVED:Dont Use

### 5.2.7.81 SPIB\_IO\_CFG Register (Offset = 150h) [Reset = 0000XXXXh]

SPIB\_IO\_CFG is shown in [Table 5-796](#).

Return to the [Summary Table](#).

**Table 5-796. SPIB\_IO\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	
18-16	miso_oen_by_cs	R/W	0h	MIBSPIB MISO OE_N Control based on Chip selectCS-applicable in slave mode 1:MISO OEN controlled based on CS.When CS is inactive OE_N=1 0:MISO OEN controlled by IP
15-11	RESERVED	R/W	0h	
10-8	cs_pol	R/W	0h	MIBSPIB CS polarity-slave mode 1: Active high 0:Active low
7-3	RESERVED	R/W	0h	
2-0	cs_deact	R/W	0h	1 : MIBSPIB External chip select is overridden with the value of MIBSPIB CS polarity-slave mode



### 5.2.7.82 SPI\_HOST\_IRQ Register (Offset = 154h) [Reset = 00000000h]

SPI\_HOST\_IRQ is shown in [Table 5-797](#).

Return to the [Summary Table](#).

**Table 5-797. SPI\_HOST\_IRQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	host_irq	R/W	0h	HOST IRQ

### 5.2.7.83 TPTC\_DBS\_CONFIG Register (Offset = 158h) [Reset = 000001XXh]

TPTC\_DBS\_CONFIG is shown in [Table 5-798](#).

Return to the [Summary Table](#).

**Table 5-798. TPTC\_DBS\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9-8	tptc_b0	R/W	1h	Default burst size tieoff value for TPTC_B0
7-6	RESERVED	R/W	0h	
5-4	tptc_a1	R/W	1h	Default burst size tieoff value for TPTC_A1
3-2	RESERVED	R/W	0h	
1-0	tptc_a0	R/W	1h	Default burst size tieoff value for TPTC_A0

### 5.2.7.84 TPCC\_PARITY\_CTRL Register (Offset = 15Ch) [Reset = 0000XXXXh]

TPCC\_PARITY\_CTRL is shown in [Table 5-799](#).

Return to the [Summary Table](#).

**Table 5-799. TPCC\_PARITY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	tpcc_a_parity_err_clr	R/W	0h	Write pulse bit field: parity clear bit. Writing 1'b1 will clear the tpcc_a_parity_addr
15-5	RESERVED	R/W	0h	
4	tpcc_a_parity_testen	R/W	0h	parity test enable for tpcc a
3-1	RESERVED	R/W	0h	
0	tpcc_a_parity_en	R/W	0h	writing 1'b1 enables parity for TPCC_A

### 5.2.7.85 TPCC\_PARITY\_STATUS Register (Offset = 160h) [Reset = 00000000h]

TPCC\_PARITY\_STATUS is shown in [Table 5-800](#).

Return to the [Summary Table](#).

**Table 5-800. TPCC\_PARITY\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	tpcc_a_parity_addr	R	0h	address where parity error happened for tpcca

### 5.2.7.86 MSS\_DBG\_ACK\_CTL0 Register (Offset = 164h) [Reset = 00XXXXXXh]

MSS\_DBG\_ACK\_CTL0 is shown in [Table 5-801](#).

Return to the [Summary Table](#).

**Table 5-801. MSS\_DBG\_ACK\_CTL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	cpsw	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
23	RESERVED	R/W	0h	
22-20	dccd	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
19	RESERVED	R/W	0h	
18-16	dccc	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
15	RESERVED	R/W	0h	
14-12	dccb	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
11	RESERVED	R/W	0h	
10-8	dcca	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
7	RESERVED	R/W	0h	
6-4	cccb	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
3	RESERVED	R/W	0h	
2-0	ccca	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor

### 5.2.7.87 MSS\_DBG\_ACK\_CTL1 Register (Offset = 168h) [Reset = 00XXXXXXh]

MSS\_DBG\_ACK\_CTL1 is shown in [Table 5-802](#).

Return to the [Summary Table](#).

**Table 5-802. MSS\_DBG\_ACK\_CTL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	scib	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
23	RESERVED	R/W	0h	
22-20	scia	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
19	RESERVED	R/W	0h	
18-16	i2c	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
15	RESERVED	R/W	0h	
14-12	mcrc	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
11	RESERVED	R/W	0h	
10-8	wdt	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
7	RESERVED	R/W	0h	
6-4	rti	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor
3	RESERVED	R/W	0h	
2-0	dcan	R/W	0h	Enable Suspend control for the peripheral. 0 :Peripheral not suspended along with processor 1: Peripehal Suspended along with procesor

### 5.2.7.88 CPSW\_CONTROL Register (Offset = 16Ch) [Reset = 0000XXXXh]

CPSW\_CONTROL is shown in [Table 5-803](#).

Return to the [Summary Table](#).

**Table 5-803. CPSW\_CONTROL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	rgmii1_id_mode	R/W	0h	writing 1'b1 would disable the internal clock delays. And those delays need to be handled on board.
15-9	RESERVED	R/W	0h	
8	rmii_ref_clk_oe_n	R/W	0h	To select the rmii_ref_clk from PAD or from MSS_RCM. 0: clock will be from mss_rcm through IO internal loopback 1: will be from
7-3	RESERVED	R/W	0h	
2-0	port1_mode_sel	R/W	0h	Port 1 Interface 00 = GMII/MII 01 = RMII 10 = RGMII 11 = Not Supported

### 5.2.7.89 MSS\_TPCC\_A\_ERRAGG\_MASK Register (Offset = 170h) [Reset = 00XXXXX0h]

MSS\_TPCC\_A\_ERRAGG\_MASK is shown in [Table 5-804](#).

Return to the [Summary Table](#).

**Table 5-804. MSS\_TPCC\_A\_ERRAGG\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	tptc_a1_read_access_error	R/W	0h	Mask Error from MSS_TPTC_A1 to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
25	tptc_a0_read_access_error	R/W	0h	Mask Error from MSS_TPTC_A0 to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
24	tpcc_a_read_access_error	R/W	0h	Mask Error from MSS_TPCC_A to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
23-19	RESERVED	R/W	0h	
18	tptc_a1_write_access_error	R/W	0h	Mask Error from MSS_TPTC_A1 to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
17	tptc_a0_write_access_error	R/W	0h	Mask Error from MSS_TPTC_A0 to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
16	tpcc_a_write_access_error	R/W	0h	Mask Error from MSS_TPCC_A to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
15-5	RESERVED	R/W	0h	
4	tpcc_a_par_err	R/W	0h	Mask Error from MSS_TPCC_A to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
3	tptc_a1_err	R/W	0h	Mask Error from MSS_TPTC_A1 to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
2	tptc_a0_err	R/W	0h	Mask Error from MSS_TPTC_A0 to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
1	tpcc_a_mpint	R/W	0h	Mask Error from MSS_TPCC_A to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked
0	tpcc_a_errint	R/W	0h	Mask Error from MSS_TPCC_A to aggregated Error MSS_TPCC_A_ERRAGG 1 : Error is Masked 0 : Error is Unmasked



### 5.2.7.90 MSS\_TPCC\_A\_ERRAGG\_STATUS Register (Offset = 174h) [Reset = 00XXXX0h]

MSS\_TPCC\_A\_ERRAGG\_STATUS is shown in [Table 5-805](#).

Return to the [Summary Table](#).

**Table 5-805. MSS\_TPCC\_A\_ERRAGG\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	tptc_a1_read_access_error	R/W	0h	Status of Error from MSS_TPTC_A1. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
25	tptc_a0_read_access_error	R/W	0h	Status of Error from MSS_TPTC_A0. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
24	tpcc_a_read_access_error	R/W	0h	Status of Error from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
23-19	RESERVED	R/W	0h	
18	tptc_a1_write_access_error	R/W	0h	Status of Error from MSS_TPTC_A1. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
17	tptc_a0_write_access_error	R/W	0h	Status of Error from MSS_TPTC_A0. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
16	tpcc_a_write_access_error	R/W	0h	Status of Error from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
15-5	RESERVED	R/W	0h	
4	tpcc_a_par_err	R/W	0h	Status of Error from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
3	tptc_a1_err	R/W	0h	Status of Error from MSS_TPTC_A1. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
2	tptc_a0_err	R/W	0h	Status of Error from MSS_TPTC_A0. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
1	tpcc_a_mpint	R/W	0h	Status of Error from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.
0	tpcc_a_errint	R/W	0h	Status of Error from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_ERRAGG_MASK Wrie 0x1 to clear this Error.

### 5.2.7.91 MSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW Register (Offset = 178h) [Reset = 00XXXXX0h]

MSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW is shown in [Table 5-806](#).

Return to the [Summary Table](#).

**Table 5-806. MSS\_TPCC\_A\_ERRAGG\_STATUS\_RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26	tptc_a1_read_access_error	R/W	0h	Raw Status of Error from MSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
25	tptc_a0_read_access_error	R/W	0h	Raw Status of Error from MSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
24	tpcc_a_read_access_error	R/W	0h	Raw Status of Error from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
23-19	RESERVED	R/W	0h	
18	tptc_a1_write_access_error	R/W	0h	Raw Status of Error from MSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
17	tptc_a0_write_access_error	R/W	0h	Raw Status of Error from MSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
16	tpcc_a_write_access_error	R/W	0h	Raw Status of Error from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
15-5	RESERVED	R/W	0h	
4	tpcc_a_par_err	R/W	0h	Raw Status of Error from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
3	tptc_a1_err	R/W	0h	Raw Status of Error from MSS_TPTC_A1. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
2	tptc_a0_err	R/W	0h	Raw Status of Error from MSS_TPTC_A0. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
1	tpcc_a_mpint	R/W	0h	Raw Status of Error from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK
0	tpcc_a_errint	R/W	0h	Raw Status of Error from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_ERRAGG_MASK

### 5.2.7.92 MSS\_TPCC\_A\_INTAGG\_MASK Register (Offset = 17Ch) [Reset = 0000XX00h]

MSS\_TPCC\_A\_INTAGG\_MASK is shown in [Table 5-807](#).

Return to the [Summary Table](#).

**Table 5-807. MSS\_TPCC\_A\_INTAGG\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	tptc_a1	R/W	0h	Mask Interrupt from TPTC A1 to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	tptc_a0	R/W	0h	Mask Interrupt from TPTC A0 to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	tpcc_a_int6	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	tpcc_a_int5	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	tpcc_a_int4	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	tpcc_a_int3	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	tpcc_a_int2	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	tpcc_a_int1	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	tpcc_a_int0	R/W	0h	Mask Interrupt from TPCC A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	tpcc_a_intg	R/W	0h	Mask Interrupt from MSS_TPCC_A to aggregated Interrupt MSS_TPCC_A_INTAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

### 5.2.7.93 MSS\_TPCC\_A\_INTAGG\_STATUS Register (Offset = 180h) [Reset = 0000XX00h]

MSS\_TPCC\_A\_INTAGG\_STATUS is shown in [Table 5-808](#).

Return to the [Summary Table](#).

**Table 5-808. MSS\_TPCC\_A\_INTAGG\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	tptc_a1	R/W	0h	Status of Interrupt from TPTC A1. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
16	tptc_a0	R/W	0h	Status of Interrupt from TPTC A0. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
7	tpcc_a_int6	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
6	tpcc_a_int5	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
5	tpcc_a_int4	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
4	tpcc_a_int3	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
3	tpcc_a_int2	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
2	tpcc_a_int1	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
1	tpcc_a_int0	R/W	0h	Status of Interrupt from TPCC A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.
0	tpcc_a_intg	R/W	0h	Status of Interrupt from MSS_TPCC_A. Set only if Interrupt is unmasked in MSS_TPCC_A_INTAGG_MASK. Write 0x1 to clear this interrupt.

### 5.2.7.94 MSS\_TPCC\_A\_INTAGG\_STATUS\_RAW Register (Offset = 184h) [Reset = 0000XX00h]

MSS\_TPCC\_A\_INTAGG\_STATUS\_RAW is shown in [Table 5-809](#).

Return to the [Summary Table](#).

**Table 5-809. MSS\_TPCC\_A\_INTAGG\_STATUS\_RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R/W	0h	
17	tptc_a1	R/W	0h	Raw Status of Interrupt from TPTC A1. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_INTAGG_MASK
16	tptc_a0	R/W	0h	Raw Status of Interrupt from TPTC A0. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_INTAGG_MASK
15-9	RESERVED	R/W	0h	
8	tpcc_a_int7	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
7	tpcc_a_int6	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
6	tpcc_a_int5	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
5	tpcc_a_int4	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
4	tpcc_a_int3	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
3	tpcc_a_int2	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
2	tpcc_a_int1	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK
1	tpcc_a_int0	R/W	0h	Raw Status of Interrupt from TPCC A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_A_INTAGG_MASK
0	tpcc_a_intg	R/W	0h	Raw Status of Interrupt from MSS_TPCC_A. Set irrespective if the Interrupt is masked or unmasked in MSS_TPCC_C_INTAGG_MASK

### 5.2.7.95 MSS\_BUS\_SAFETY\_CTRL Register (Offset = 1A0h) [Reset = 00000000h]

MSS\_BUS\_SAFETY\_CTRL is shown in [Table 5-810](#).

Return to the [Summary Table](#).

**Table 5-810. MSS\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	enable	R/W	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.96 MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_CTRL Register (Offset = 1A4h) [Reset = 001FXXXXh]

MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_CTRL is shown in [Table 5-811](#).

Return to the [Summary Table](#).

**Table 5-811. MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	1Fh	Refer to TPR12 Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.97 MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_FI Register (Offset = 1A8h) [Reset = 000000X0h]

MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_FI is shown in [Table 5-812](#).

Return to the [Summary Table](#).

**Table 5-812. MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
23-16	main	R/W	0h	Refer to TPR12 Substem Microarch document for more details
15-8	data	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to TPR12 Substem Microarch document for more details
4	sec	R/W	0h	Refer to TPR12 Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
0	global_main	R/W	0h	Refer to TPR12 Substem Microarch document for more details



### 5.2.7.98 MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR Register (Offset = 1ACh) [Reset = 00000000h]

MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR is shown in [Table 5-813](#).

Return to the [Summary Table](#).

**Table 5-813. MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to TPR12 Substem Microarch document for more details
23-16	sec	R	0h	Refer to TPR12 Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.99 MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 1B0h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-814](#).

Return to the [Summary Table](#).

**Table 5-814. MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	d0	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.100 MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 1B4h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-815](#).

Return to the [Summary Table](#).

**Table 5-815. MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.101 MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 1B8h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-816](#).

Return to the [Summary Table](#).

**Table 5-816. MSS\_CR5A\_AXI\_RD\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.102 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_CTRL Register (Offset = 1D4h) [Reset = 001FXXXh]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_CTRL is shown in [Table 5-817](#).

Return to the [Summary Table](#).

**Table 5-817. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	1Fh	Refer to TPR12 Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.103 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_FI Register (Offset = 1D8h) [Reset = 000000X0h]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_FI is shown in [Table 5-818](#).

Return to the [Summary Table](#).

**Table 5-818. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
23-16	main	R/W	0h	Refer to TPR12 Substem Microarch document for more details
15-8	data	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to TPR12 Substem Microarch document for more details
4	sec	R/W	0h	Refer to TPR12 Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
0	global_main	R/W	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.104 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR Register (Offset = 1DCh) [Reset = 00000000h]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR is shown in [Table 5-819](#).

Return to the [Summary Table](#).

**Table 5-819. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to TPR12 Substem Microarch document for more details
23-16	sec	R	0h	Refer to TPR12 Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.105 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 1E0h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-820](#).

Return to the [Summary Table](#).

**Table 5-820. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	d0	R	0h	Refer to TPR12 Substem Microarch document for more details



### 5.2.7.106 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 1E4h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-821](#).

Return to the [Summary Table](#).

**Table 5-821. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.107 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 1E8h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-822](#).

Return to the [Summary Table](#).

**Table 5-822. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.108 MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 1ECh) [Reset = 00000000h]

MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-823](#).

Return to the [Summary Table](#).

**Table 5-823. MSS\_CR5A\_AXI\_WR\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.109 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_CTRL Register (Offset = 20Ch) [Reset = 001FXXXh]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_CTRL is shown in [Table 5-824](#).

Return to the [Summary Table](#).

**Table 5-824. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	1Fh	Refer to TPR12 Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.110 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_FI Register (Offset = 210h) [Reset = 000000X0h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_FI is shown in [Table 5-825](#).

Return to the [Summary Table](#).

**Table 5-825. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
23-16	main	R/W	0h	Refer to TPR12 Substem Microarch document for more details
15-8	data	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to TPR12 Substem Microarch document for more details
4	sec	R/W	0h	Refer to TPR12 Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
0	global_main	R/W	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.111 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR Register (Offset = 214h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR is shown in [Table 5-826](#).

Return to the [Summary Table](#).

**Table 5-826. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to TPR12 Substem Microarch document for more details
23-16	sec	R	0h	Refer to TPR12 Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.112 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 218h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-827](#).

Return to the [Summary Table](#).

**Table 5-827. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	d0	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.113 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 21Ch) [Reset = 00000000h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-828](#).

Return to the [Summary Table](#).

**Table 5-828. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details



### 5.2.7.114 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 220h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-829](#).

Return to the [Summary Table](#).

**Table 5-829. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.115 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 224h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-830](#).

Return to the [Summary Table](#).

**Table 5-830. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.116 MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 228h) [Reset = 00000000h]

MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-831](#).

Return to the [Summary Table](#).

**Table 5-831. MSS\_CR5A\_AXI\_S\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.117 MSS\_L2\_A\_BUS\_SAFETY\_CTRL Register (Offset = 470h) [Reset = 001FXXXh]

MSS\_L2\_A\_BUS\_SAFETY\_CTRL is shown in [Table 5-832](#).

Return to the [Summary Table](#).

**Table 5-832. MSS\_L2\_A\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	1Fh	Refer to TPR12 Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.118 MSS\_L2\_A\_BUS\_SAFETY\_FI Register (Offset = 474h) [Reset = 000000X0h]

MSS\_L2\_A\_BUS\_SAFETY\_FI is shown in [Table 5-833](#).

Return to the [Summary Table](#).

**Table 5-833. MSS\_L2\_A\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
23-16	main	R/W	0h	Refer to TPR12 Substem Microarch document for more details
15-8	data	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to TPR12 Substem Microarch document for more details
4	sec	R/W	0h	Refer to TPR12 Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
0	global_main	R/W	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.119 MSS\_L2\_A\_BUS\_SAFETY\_ERR Register (Offset = 478h) [Reset = 00000000h]

MSS\_L2\_A\_BUS\_SAFETY\_ERR is shown in [Table 5-834](#).

Return to the [Summary Table](#).

**Table 5-834. MSS\_L2\_A\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to TPR12 Substem Microarch document for more details
23-16	sec	R	0h	Refer to TPR12 Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.120 MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 47Ch) [Reset = 00000000h]

MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-835](#).

Return to the [Summary Table](#).

**Table 5-835. MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	d0	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.121 MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 480h) [Reset = 00000000h]

MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-836](#).

Return to the [Summary Table](#).

**Table 5-836. MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details



### 5.2.7.122 MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 484h) [Reset = 00000000h]

MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-837](#).

Return to the [Summary Table](#).

**Table 5-837. MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.123 MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 488h) [Reset = 00000000h]

MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-838](#).

Return to the [Summary Table](#).

**Table 5-838. MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.124 MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 48Ch) [Reset = 00000000h]

MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-839](#).

Return to the [Summary Table](#).

**Table 5-839. MSS\_L2\_A\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.125 MSS\_L2\_B\_BUS\_SAFETY\_CTRL Register (Offset = 490h) [Reset = 001FXXXh]

MSS\_L2\_B\_BUS\_SAFETY\_CTRL is shown in [Table 5-840](#).

Return to the [Summary Table](#).

**Table 5-840. MSS\_L2\_B\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	1Fh	Refer to TPR12 Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.126 MSS\_L2\_B\_BUS\_SAFETY\_FI Register (Offset = 494h) [Reset = 000000X0h]

MSS\_L2\_B\_BUS\_SAFETY\_FI is shown in [Table 5-841](#).

Return to the [Summary Table](#).

**Table 5-841. MSS\_L2\_B\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
23-16	main	R/W	0h	Refer to TPR12 Substem Microarch document for more details
15-8	data	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to TPR12 Substem Microarch document for more details
4	sec	R/W	0h	Refer to TPR12 Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
0	global_main	R/W	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.127 MSS\_L2\_B\_BUS\_SAFETY\_ERR Register (Offset = 498h) [Reset = 00000000h]

MSS\_L2\_B\_BUS\_SAFETY\_ERR is shown in [Table 5-842](#).

Return to the [Summary Table](#).

**Table 5-842. MSS\_L2\_B\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to TPR12 Substem Microarch document for more details
23-16	sec	R	0h	Refer to TPR12 Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.128 MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 49Ch) [Reset = 00000000h]

MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-843](#).

Return to the [Summary Table](#).

**Table 5-843. MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	d0	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.129 MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 4A0h) [Reset = 00000000h]

MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-844](#).

Return to the [Summary Table](#).

**Table 5-844. MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details



### 5.2.7.130 MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 4A4h) [Reset = 00000000h]

MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-845](#).

Return to the [Summary Table](#).

**Table 5-845. MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.131 MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 4A8h) [Reset = 00000000h]

MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-846](#).

Return to the [Summary Table](#).

**Table 5-846. MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.132 MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 4ACh) [Reset = 00000000h]

MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-847](#).

Return to the [Summary Table](#).

**Table 5-847. MSS\_L2\_B\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.133 MSS\_BUS\_SAFETY\_SEC\_ERR\_STAT0 Register (Offset = 510h) [Reset = 0XXXXXXXh]

MSS\_BUS\_SAFETY\_SEC\_ERR\_STAT0 is shown in [Table 5-848](#).

Return to the [Summary Table](#).

**Table 5-848. MSS\_BUS\_SAFETY\_SEC\_ERR\_STAT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	
26	l2ram1	R	0h	Bus safety single-bit-error of Node mentioned in the field
25	l2ram0	R	0h	Bus safety single-bit-error of Node mentioned in the field
24-5	RESERVED	R	0h	
4	cr5a_slv	R	0h	Bus safety single-bit-error of Node mentioned in the field
3	RESERVED	R	0h	
2	cr5a_wr	R	0h	Bus safety single-bit-error of Node mentioned in the field
1	RESERVED	R	0h	
0	cr5a_rd	R	0h	Bus safety single-bit-error of Node mentioned in the field

### 5.2.7.134 MSS\_BUS\_SAFETY\_SEC\_ERR\_STAT1 Register (Offset = 514h) [Reset = 00000000h]

MSS\_BUS\_SAFETY\_SEC\_ERR\_STAT1 is shown in [Table 5-849](#).

Return to the [Summary Table](#).

**Table 5-849. MSS\_BUS\_SAFETY\_SEC\_ERR\_STAT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4	l2ram2	R	0h	Bus safety single-bit-error of Node mentioned in the field
3	r5ss2dss	R	0h	Bus safety single-bit-error of Node mentioned in the field
2	dss2r5ss	R	0h	Bus safety single-bit-error of Node mentioned in the field
1	r5ss2mss	R	0h	Bus safety single-bit-error of Node mentioned in the field
0	mss2r5ss	R	0h	Bus safety single-bit-error of Node mentioned in the field

### 5.2.7.135 HW\_REG0 Register (Offset = 518h) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-850](#).

Return to the [Summary Table](#).

**Table 5-850. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg0	R/W	0h	HW reserved Register

### 5.2.7.136 HW\_REG1 Register (Offset = 51Ch) [Reset = 00000000h]

HW\_REG1 is shown in [Table 5-851](#).

Return to the [Summary Table](#).

**Table 5-851. HW\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg1	R/W	0h	HW reserved Regsiter

### 5.2.7.137 PREVIOUS\_NAME Register (Offset = 520h) [Reset = 00XXXXXXh]

PREVIOUS\_NAME is shown in [Table 5-852](#).

Return to the [Summary Table](#).

**Table 5-852. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	hwreg2	R/W	0h	HW reserved Register
24	dbg_pp_switch_en	R/W	0h	Writing 1'b1 will allow CPSW ping-pong switch to happen during debugger reads (through JTAG).
23-21	RESERVED	R/W	0h	
20	pp_switch_ovr	R/W	0h	Writing 1'b1 takes override on the CPWS ping-pong switch. Ping-pong switch doesn't happen even after reading the configured end-address.
19-17	RESERVED	R/W	0h	
16	CPSW_CLK_STOP_REQ	R/W	0h	CPSW clock stop req
15-13	RESERVED	R/W	0h	
12	set	R/W	0h	Write pulse bit field: Write 1'b1 whenever ping_pong_sel needs to be set.
11-9	RESERVED	R/W	0h	
8	clr	R/W	0h	Write pulse bit field: Write 1'b1 whenever ping_pong_sel needs to be cleared.
7-5	RESERVED	R/W	0h	
4	crc_enable	R/W	0h	1'b0: CRC disable 1'b1: CRC enable
3-1	RESERVED	R/W	0h	
0	crc_mode	R/W	0h	1'b0 : CRC16CCITT 1'b1 : ETHERNET CRC32



### 5.2.7.138 HW\_REG3 Register (Offset = 524h) [Reset = 000001XXh]

HW\_REG3 is shown in [Table 5-853](#).

Return to the [Summary Table](#).

**Table 5-853. HW\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	hwreg3	R	0h	HW reserved Register
8	CPSW_CLK_STOP_IDLE	R	1h	CPSW clock stop idle
7-5	RESERVED	R	0h	
4	CPSW_CLK_STOP_ACK	R	0h	CPSW clock stop ack
3-1	RESERVED	R	0h	
0	ro	R	0h	This denotes the status of ping_pong_sel bit.

### 5.2.7.139 HW\_REG4 Register (Offset = 528h) [Reset = 000000FFh]

HW\_REG4 is shown in [Table 5-854](#).

Return to the [Summary Table](#).

**Table 5-854. HW\_REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	mss_pcrA_timeout_cfg	R/W	FFh	Configures the timeout value for MSS_PCRA.

### 5.2.7.140 HW\_REG5 Register (Offset = 52Ch) [Reset = 00000000h]

HW\_REG5 is shown in [Table 5-855](#).

Return to the [Summary Table](#).

**Table 5-855. HW\_REG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg5	R/W	0h	HW reserved Register

### 5.2.7.141 HW\_REG6 Register (Offset = 530h) [Reset = 00000000h]

HW\_REG6 is shown in [Table 5-856](#).

Return to the [Summary Table](#).

**Table 5-856. HW\_REG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg6	R/W	0h	HW reserved Register

### 5.2.7.142 HW\_REG7 Register (Offset = 534h) [Reset = 00000000h]

HW\_REG7 is shown in [Table 5-857](#).

Return to the [Summary Table](#).

**Table 5-857. HW\_REG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hwreg7	R/W	0h	HW reserved Register

### 5.2.7.143 MSS\_CR5A\_AHB\_BUS\_SAFETY\_CTRL Register (Offset = 5B8h) [Reset = 0000XXXXh]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_CTRL is shown in [Table 5-858](#).

Return to the [Summary Table](#).

**Table 5-858. MSS\_CR5A\_AHB\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	0h	Refer to TPR12 Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.144 MSS\_CR5A\_AHB\_BUS\_SAFETY\_FI Register (Offset = 5BCh) [Reset = 000000X0h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_FI is shown in [Table 5-859](#).

Return to the [Summary Table](#).

**Table 5-859. MSS\_CR5A\_AHB\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
23-16	main	R/W	0h	Refer to TPR12 Substem Microarch document for more details
15-8	data	R/W	0h	Refer to TPR12 Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to TPR12 Substem Microarch document for more details
4	sec	R/W	0h	Refer to TPR12 Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to TPR12 Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to TPR12 Substem Microarch document for more details
0	global_main	R/W	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.145 MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR Register (Offset = 5C0h) [Reset = 00000000h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR is shown in [Table 5-860](#).

Return to the [Summary Table](#).

**Table 5-860. MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to TPR12 Substem Microarch document for more details
23-16	sec	R	0h	Refer to TPR12 Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to TPR12 Substem Microarch document for more details



### 5.2.7.146 MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 5C4h) [Reset = 00000000h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-861](#).

Return to the [Summary Table](#).

**Table 5-861. MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to TPR12 Substem Microarch document for more details
7-0	d0	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.147 MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 5C8h) [Reset = 00000000h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-862](#).

Return to the [Summary Table](#).

**Table 5-862. MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.148 MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 5CCh) [Reset = 00000000h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-863](#).

Return to the [Summary Table](#).

**Table 5-863. MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.149 MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 5D0h) [Reset = 00000000h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-864](#).

Return to the [Summary Table](#).

**Table 5-864. MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.150 MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 5D4h) [Reset = 00000000h]

MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-865](#).

Return to the [Summary Table](#).

**Table 5-865. MSS\_CR5A\_AHB\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to TPR12 Substem Microarch document for more details

### 5.2.7.151 DMM\_CTRL\_REG Register (Offset = 5F8h) [Reset = 00000000h]

DMM\_CTRL\_REG is shown in [Table 5-866](#).

Return to the [Summary Table](#).

**Table 5-866. DMM\_CTRL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	dmm_pad_select	R/W	0h	0: SOC will be able to send the packet to DMMA/B 1: PAD will be able to send the packet to DMMA/B controlling from PAD

### 5.2.7.152 MSS\_CR5A\_MBOX\_WRITE\_DONE Register (Offset = 5FCh) [Reset = 0XXXXXXXh]

MSS\_CR5A\_MBOX\_WRITE\_DONE is shown in [Table 5-867](#).

Return to the [Summary Table](#).

**Table 5-867. MSS\_CR5A\_MBOX\_WRITE\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	proc_7	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 7
27-25	RESERVED	R/W	0h	
24	proc_6	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 6
23-21	RESERVED	R/W	0h	
20	proc_5	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 5
19-17	RESERVED	R/W	0h	
16	proc_4	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 4
15-13	RESERVED	R/W	0h	
12	proc_3	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 3
11-9	RESERVED	R/W	0h	
8	proc_2	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 2
7-5	RESERVED	R/W	0h	
4	proc_1	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 1
3-1	RESERVED	R/W	0h	
0	proc_0	R/W	0h	Write pulse bit field: This register should be written once finishing writing into the mailbox memory of processor 0

### 5.2.7.153 MSS\_CR5A\_MBOX\_READ\_REQ Register (Offset = 600h) [Reset = 0XXXXXXXh]

MSS\_CR5A\_MBOX\_READ\_REQ is shown in [Table 5-868](#).

Return to the [Summary Table](#).

**Table 5-868. MSS\_CR5A\_MBOX\_READ\_REQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	proc_7	R/W	0h	This is request from processor 7 to mss_cr5a. Requesting it to read from mailbox.
27-25	RESERVED	R/W	0h	
24	proc_6	R/W	0h	This is request from processor 6 to mss_cr5a. Requesting it to read from mailbox.
23-21	RESERVED	R/W	0h	
20	proc_5	R/W	0h	This is request from processor 5 to mss_cr5a. Requesting it to read from mailbox.
19-17	RESERVED	R/W	0h	
16	proc_4	R/W	0h	This is request from processor 4 to mss_cr5a. Requesting it to read from mailbox.
15-13	RESERVED	R/W	0h	
12	proc_3	R/W	0h	This is request from processor 3 to mss_cr5a. Requesting it to read from mailbox.
11-9	RESERVED	R/W	0h	
8	proc_2	R/W	0h	This is request from processor 2 to mss_cr5a. Requesting it to read from mailbox.
7-5	RESERVED	R/W	0h	
4	proc_1	R/W	0h	This is request from processor 1 to mss_cr5a. Requesting it to read from mailbox.
3-1	RESERVED	R/W	0h	
0	proc_0	R/W	0h	This is request from processor 0 to mss_cr5a. Requesting it to read from mailbox.



### 5.2.7.154 MSS\_CR5A\_MBOX\_READ\_DONE Register (Offset = 604h) [Reset = 0XXXXXXXh]

MSS\_CR5A\_MBOX\_READ\_DONE is shown in [Table 5-869](#).

Return to the [Summary Table](#).

**Table 5-869. MSS\_CR5A\_MBOX\_READ\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	proc_7	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 7
27-25	RESERVED	R/W	0h	
24	proc_6	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 6
23-21	RESERVED	R/W	0h	
20	proc_5	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 5
19-17	RESERVED	R/W	0h	
16	proc_4	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 4
15-13	RESERVED	R/W	0h	
12	proc_3	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 3
11-9	RESERVED	R/W	0h	
8	proc_2	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 2
7-5	RESERVED	R/W	0h	
4	proc_1	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 1
3-1	RESERVED	R/W	0h	
0	proc_0	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 0

### 5.2.7.155 MSS\_PBIST\_KEY\_RST Register (Offset = 614h) [Reset = 00000000h]

MSS\_PBIST\_KEY\_RST is shown in [Table 5-870](#).

Return to the [Summary Table](#).

**Table 5-870. MSS\_PBIST\_KEY\_RST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-4	pbist_st_rst	R/W	0h	MSS PBIST controller will be brought out of reset when value is 0xA
3-0	pbist_st_key	R/W	0h	Top PBIST Selftest Key. Valid value is 0x5

### 5.2.7.156 MSS\_PBIST\_REG0 Register (Offset = 618h) [Reset = 00000000h]

MSS\_PBIST\_REG0 is shown in [Table 5-871](#).

Return to the [Summary Table](#).

**Table 5-871. MSS\_PBIST\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	pbist_reg	R/W	0h	

### 5.2.7.157 MSS\_PBIST\_REG1 Register (Offset = 61Ch) [Reset = 00000000h]

MSS\_PBIST\_REG1 is shown in [Table 5-872](#).

Return to the [Summary Table](#).

**Table 5-872. MSS\_PBIST\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	pbist_reg	R/W	0h	

### 5.2.7.158 MSS\_PBIST\_REG2 Register (Offset = 620h) [Reset = 00000000h]

MSS\_PBIST\_REG2 is shown in [Table 5-873](#).

Return to the [Summary Table](#).

**Table 5-873. MSS\_PBIST\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	pbist_reg	R/W	0h	

### 5.2.7.159 MSS\_QSPI\_CONFIG Register (Offset = 624h) [Reset = 000000XXh]

MSS\_QSPI\_CONFIG is shown in [Table 5-874](#).

Return to the [Summary Table](#).

**Table 5-874. MSS\_QSPI\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	clk_loopback	R/W	0h	Write 3'b111 to take board level loop back clock for QSPI
7-3	RESERVED	R/W	0h	
2-0	ext_clk	R/W	0h	Write 3'b111 to external clock as QSPI baud clock source needed for DFT IO char.

### 5.2.7.160 MSS\_STC\_CONTROL Register (Offset = 628h) [Reset = 00000000h]

MSS\_STC\_CONTROL is shown in [Table 5-875](#).

Return to the [Summary Table](#).

**Table 5-875. MSS\_STC\_CONTROL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	cr5_wfi_override	R/W	0h	writing 3'b111 will bypass the wfi signals from R5SS.

### 5.2.7.161 MSS\_CTI\_TRIG\_SEL Register (Offset = 62Ch) [Reset = 00000000h]

MSS\_CTI\_TRIG\_SEL is shown in [Table 5-876](#).

Return to the [Summary Table](#).

**Table 5-876. MSS\_CTI\_TRIG\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	trig8_sel	R/W	0h	Used for selecting the trigger source for 8th trigger of MSS_CTI



### 5.2.7.162 MSS\_DBGSS\_CTI\_TRIG\_SEL Register (Offset = 630h) [Reset = 00000000h]

MSS\_DBGSS\_CTI\_TRIG\_SEL is shown in [Table 5-877](#).

Return to the [Summary Table](#).

**Table 5-877. MSS\_DBGSS\_CTI\_TRIG\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	trig3	R/W	0h	Used for selecting the trigger source for 3rd trigger of ONE_MCU_CTI
15-8	trig2	R/W	0h	Used for selecting the trigger source for 2nd trigger of ONE_MCU_CTI
7-0	trig1	R/W	0h	Used for selecting the trigger source for 1st trigger of ONE_MCU_CTI

### 5.2.7.163 MSS\_BOOT\_INFO\_REG0 Register (Offset = 634h) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG0 is shown in [Table 5-878](#).

Return to the [Summary Table](#).

**Table 5-878. MSS\_BOOT\_INFO\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.164 MSS\_BOOT\_INFO\_REG1 Register (Offset = 638h) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG1 is shown in [Table 5-879](#).

Return to the [Summary Table](#).

**Table 5-879. MSS\_BOOT\_INFO\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.165 MSS\_BOOT\_INFO\_REG2 Register (Offset = 63Ch) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG2 is shown in [Table 5-880](#).

Return to the [Summary Table](#).

**Table 5-880. MSS\_BOOT\_INFO\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.166 MSS\_BOOT\_INFO\_REG3 Register (Offset = 640h) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG3 is shown in [Table 5-881](#).

Return to the [Summary Table](#).

**Table 5-881. MSS\_BOOT\_INFO\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.167 MSS\_BOOT\_INFO\_REG4 Register (Offset = 644h) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG4 is shown in [Table 5-882](#).

Return to the [Summary Table](#).

**Table 5-882. MSS\_BOOT\_INFO\_REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.168 MSS\_BOOT\_INFO\_REG5 Register (Offset = 648h) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG5 is shown in [Table 5-883](#).

Return to the [Summary Table](#).

**Table 5-883. MSS\_BOOT\_INFO\_REG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.169 MSS\_BOOT\_INFO\_REG6 Register (Offset = 64Ch) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG6 is shown in [Table 5-884](#).

Return to the [Summary Table](#).

**Table 5-884. MSS\_BOOT\_INFO\_REG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use



### 5.2.7.170 MSS\_BOOT\_INFO\_REG7 Register (Offset = 650h) [Reset = 00000000h]

MSS\_BOOT\_INFO\_REG7 is shown in [Table 5-885](#).

Return to the [Summary Table](#).

**Table 5-885. MSS\_BOOT\_INFO\_REG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.7.171 MSS\_TPTC\_ECCAGGR\_CLK\_CNTRL Register (Offset = 654h) [Reset = 00000007h]

MSS\_TPTC\_ECCAGGR\_CLK\_CNTRL is shown in [Table 5-886](#).

Return to the [Summary Table](#).

**Table 5-886. MSS\_TPTC\_ECCAGGR\_CLK\_CNTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2	tptc_B0	R/W	1h	Writing '0' will gate the clock to TPTC_B0-FIFO during ECC-AGGR interaction(fault injection)
1	tptc_A1	R/W	1h	Writing '0' will gate the clock to TPTC_A1-FIFO during ECC-AGGR interaction(fault injection)
0	tptc_A0	R/W	1h	Writing '0' will gate the clock to TPTC_A0-FIFO during ECC-AGGR interaction(fault injection)

### 5.2.7.172 MSS\_PERIPH\_ERRAGG\_MASK0 Register (Offset = 658h) [Reset = 00000000h]

MSS\_PERIPH\_ERRAGG\_MASK0 is shown in [Table 5-887](#).

Return to the [Summary Table](#).

**Table 5-887. MSS\_PERIPH\_ERRAGG\_MASK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27	top_mdo_wr	R/W	0h	RESERVED:Dont Use
26	top_mdo_rd	R/W	0h	RESERVED:Dont Use
25	rcss_rcm_wr	R/W	0h	Mask Interrupt from RSS_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
24	rcss_rcm_rd	R/W	0h	Mask Interrupt from RSS_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
23	rcss_ctrl_wr	R/W	0h	Mask Interrupt from RSS_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
22	rcss_ctrl_rd	R/W	0h	Mask Interrupt from RSS_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
21	hwa_cfg_wr	R/W	0h	Mask Interrupt from HWA_CFG to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
20	hwa_cfg_rd	R/W	0h	Mask Interrupt from HWA_CFG to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
19	dss_cm4_ctrl_wr	R/W	0h	RESERVED:Dont Use
18	dss_cm4_ctrl_rd	R/W	0h	RESERVED:Dont Use
17	dss_rcm_wr	R/W	0h	Mask Interrupt from DSS_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
16	dss_rcm_rd	R/W	0h	Mask Interrupt from DSS_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15	dss_ctrl_wr	R/W	0h	Mask Interrupt from DSS_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
14	dss_ctrl_rd	R/W	0h	Mask Interrupt from DSS_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
13	hsm_ctrl_wr	R/W	0h	Mask Interrupt from HSM_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
12	hsm_ctrl_rd	R/W	0h	Mask Interrupt from HSM_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
11	hsm_soc_ctrl_wr	R/W	0h	Mask Interrupt from HSM_SOC_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
10	hsm_soc_ctrl_rd	R/W	0h	Mask Interrupt from HSM_SOC_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
9	top_aurora_wr	R/W	0h	RESERVED:Dont Use
8	top_aurora_rd	R/W	0h	RESERVED:Dont Use

**Table 5-887. MSS\_PERIPH\_ERRAGG\_MASK0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7	top_rcm_wr	R/W	0h	Mask Interrupt from TOP_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	top_rcm_rd	R/W	0h	Mask Interrupt from TOP_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	top_ctrl_wr	R/W	0h	Mask Interrupt from TOP_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	top_ctrl_rd	R/W	0h	Mask Interrupt from TOP_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	mss_rcm_wr	R/W	0h	Mask Interrupt from MSS_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	mss_rcm_rd	R/W	0h	Mask Interrupt from MSS_RCM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	mss_ctrl_wr	R/W	0h	Mask Interrupt from MSS_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	mss_ctrl_rd	R/W	0h	Mask Interrupt from MSS_CTRL to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

### 5.2.7.173 MSS\_PERIPH\_ERRAGG\_STATUS0 Register (Offset = 65Ch) [Reset = 00000000h]

MSS\_PERIPH\_ERRAGG\_STATUS0 is shown in [Table 5-888](#).

Return to the [Summary Table](#).

**Table 5-888. MSS\_PERIPH\_ERRAGG\_STATUS0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27	top_mdo_wr	R/W	0h	RESERVED:Dont Use
26	top_mdo_rd	R/W	0h	RESERVED:Dont Use
25	rcss_rcm_wr	R/W	0h	Status of Interrupt from RSS_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
24	rcss_rcm_rd	R/W	0h	Status of Interrupt from RSS_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
23	rcss_ctrl_wr	R/W	0h	Status of Interrupt from RSS_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
22	rcss_ctrl_rd	R/W	0h	Status of Interrupt from RSS_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
21	hwa_cfg_wr	R/W	0h	Status of Interrupt from HWA_CFG Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
20	hwa_cfg_rd	R/W	0h	Status of Interrupt from HWA_CFG Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
19	dss_cm4_ctrl_wr	R/W	0h	RESERVED:Dont Use
18	dss_cm4_ctrl_rd	R/W	0h	RESERVED:Dont Use
17	dss_rcm_wr	R/W	0h	Status of Interrupt from DSS_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
16	dss_rcm_rd	R/W	0h	Status of Interrupt from DSS_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
15	dss_ctrl_wr	R/W	0h	Status of Interrupt from DSS_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
14	dss_ctrl_rd	R/W	0h	Status of Interrupt from DSS_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
13	hsm_ctrl_wr	R/W	0h	Status of Interrupt from HSM_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
12	hsm_ctrl_rd	R/W	0h	Status of Interrupt from HSM_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
11	hsm_soc_ctrl_wr	R/W	0h	Status of Interrupt from HSM_SOC_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
10	hsm_soc_ctrl_rd	R/W	0h	Status of Interrupt from HSM_SOC_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
9	top_aurora_wr	R/W	0h	RESERVED:Dont Use
8	top_aurora_rd	R/W	0h	RESERVED:Dont Use
7	top_rcm_wr	R/W	0h	Status of Interrupt from TOP_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
6	top_rcm_rd	R/W	0h	Status of Interrupt from TOP_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
5	top_ctrl_wr	R/W	0h	Status of Interrupt from TOP_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
4	top_ctrl_rd	R/W	0h	Status of Interrupt from TOP_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
3	mss_rcm_wr	R/W	0h	Status of Interrupt from MSS_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.

**Table 5-888. MSS\_PERIPH\_ERRAGG\_STATUS0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	mss_rcm_rd	R/W	0h	Status of Interrupt from MSS_RCM Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
1	mss_ctrl_wr	R/W	0h	Status of Interrupt from MSS_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.
0	mss_ctrl_rd	R/W	0h	Status of Interrupt from MSS_CTRL Set only if Interupt is unmasked in MSS_PERIPH_ERRAGG_MASK0 Wrie 0x1 to clear this interrupt.

### 5.2.7.174 MSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 Register (Offset = 660h) [Reset = 00000000h]

MSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 is shown in [Table 5-889](#).

Return to the [Summary Table](#).

**Table 5-889. MSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27	top_mdo_wr	R/W	0h	RESERVED:Dont Use
26	top_mdo_rd	R/W	0h	RESERVED:Dont Use
25	rcss_rcm_wr	R/W	0h	Raw Status of Interrupt from RSS_RCM. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
24	rcss_rcm_rd	R/W	0h	Raw Status of Interrupt from RSS_RCM. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
23	rcss_ctrl_wr	R/W	0h	Raw Status of Interrupt from RSS_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
22	rcss_ctrl_rd	R/W	0h	Raw Status of Interrupt from RSS_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
21	hwa_cfg_wr	R/W	0h	Raw Status of Interrupt from HWA_CFG. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
20	hwa_cfg_rd	R/W	0h	Raw Status of Interrupt from HWA_CFG. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
19	dss_cm4_ctrl_wr	R/W	0h	RESERVED:Dont Use
18	dss_cm4_ctrl_rd	R/W	0h	RESERVED:Dont Use
17	dss_rcm_wr	R/W	0h	Raw Status of Interrupt from DSS_RCM. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
16	dss_rcm_rd	R/W	0h	Raw Status of Interrupt from DSS_RCM. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
15	dss_ctrl_wr	R/W	0h	Raw Status of Interrupt from DSS_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
14	dss_ctrl_rd	R/W	0h	Raw Status of Interrupt from DSS_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
13	hsm_ctrl_wr	R/W	0h	Raw Status of Interrupt from HSM_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
12	hsm_ctrl_rd	R/W	0h	Raw Status of Interrupt from HSM_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
11	hsm_soc_ctrl_wr	R/W	0h	Raw Status of Interrupt from HSM_SOC_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
10	hsm_soc_ctrl_rd	R/W	0h	Raw Status of Interrupt from HSM_SOC_CTRL. Set irrespective if the Interupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
9	top_aurora_wr	R/W	0h	RESERVED:Dont Use
8	top_aurora_rd	R/W	0h	RESERVED:Dont Use

**Table 5-889. MSS\_PERIPH\_ERRAGG\_STATUS\_RAW0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7	top_rcm_wr	R/W	0h	Raw Status of Interrupt from TOP_RCM. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
6	top_rcm_rd	R/W	0h	Raw Status of Interrupt from TOP_RCM. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
5	top_ctrl_wr	R/W	0h	Raw Status of Interrupt from TOP_CTRL. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
4	top_ctrl_rd	R/W	0h	Raw Status of Interrupt from TOP_CTRL. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
3	mss_rcm_wr	R/W	0h	Raw Status of Interrupt from MSS_RCM. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
2	mss_rcm_rd	R/W	0h	Raw Status of Interrupt from MSS_RCM. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
1	mss_ctrl_wr	R/W	0h	Raw Status of Interrupt from MSS_CTRL. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0
0	mss_ctrl_rd	R/W	0h	Raw Status of Interrupt from MSS_CTRL. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK0



### 5.2.7.175 MSS\_PERIPH\_ERRAGG\_MASK1 Register (Offset = 664h) [Reset = 00000000h]

MSS\_PERIPH\_ERRAGG\_MASK1 is shown in [Table 5-890](#).

Return to the [Summary Table](#).

**Table 5-890. MSS\_PERIPH\_ERRAGG\_MASK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	mpu_rd_hsm	R/W	0h	Mask Interrupt from MPU_DSS_HSM to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
15	mpu_rd_dss_mbox	R/W	0h	Mask Interrupt from MPU_DSS_MBOX to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
14	mpu_rd_dss_hwa_proc	R/W	0h	RESERVED:Dont Use
13	mpu_rd_dss_hwa_dma1	R/W	0h	MRESERVED:Dont Use
12	mpu_rd_dss_hwa_dma0	R/W	0h	RESERVED:Dont Use
11	mpu_rd_dss_l3_bankd	R/W	0h	RESERVED:Dont Use
10	mpu_rd_dss_l3_bankc	R/W	0h	RESERVED:Dont Use
9	mpu_rd_dss_l3_bankb	R/W	0h	Mask Interrupt from MPU_DSS_L3_BANKB to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
8	mpu_rd_dss_l3_banka	R/W	0h	Mask Interrupt from MPU_DSS_L3_BANKA to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
7	mpu_rd_mss_cr5b_axis	R/W	0h	Mask Interrupt from MPU_MSS_L2_BANKC to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
6	mpu_rd_mss_cr5a_axis	R/W	0h	Mask Interrupt from MPU_MSS_CR5A_AXIS to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
5	mpu_rd_mss_qspi	R/W	0h	Mask Interrupt from MPU_MSS_QSPI to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
4	mpu_rd_mss_pcra	R/W	0h	Mask Interrupt from MPU_MSS_PCRA to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
3	mpu_rd_mss_mbox	R/W	0h	Mask Interrupt from MPU_MSS_MBOX to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
2	mpu_rd_hsm_dthe	R/W	0h	Mask Interrupt from MPU_HSM_DTHE to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
1	mpu_rd_mss_l2_bankb	R/W	0h	Mask Interrupt from MPU_MSS_L2_BANKB to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked
0	mpu_rd_mss_l2_banka	R/W	0h	Mask Interrupt from MPU_MSS_L2_BANKA to aggregated Interrupt MSS_PERIPH_ACCESS_ERRAGG 1 : Interrupt is Masked 0 : Interrupt is Unmasked

### 5.2.7.176 MSS\_PERIPH\_ERRAGG\_STATUS1 Register (Offset = 668h) [Reset = 00000000h]

MSS\_PERIPH\_ERRAGG\_STATUS1 is shown in [Table 5-891](#).

Return to the [Summary Table](#).

**Table 5-891. MSS\_PERIPH\_ERRAGG\_STATUS1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	mpu_rd_hsm	R/W	0h	Status of Interrupt from MPU_HSM Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
15	mpu_rd_dss_mbox	R/W	0h	Status of Interrupt from MPU_DSS_MBOX Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
14	mpu_rd_dss_hwa_proc	R/W	0h	RESERVED: Dont Use
13	mpu_rd_dss_hwa_dma1	R/W	0h	RESERVED: Dont Use
12	mpu_rd_dss_hwa_dma0	R/W	0h	RESERVED: Dont Use
11	mpu_rd_dss_l3_bankd	R/W	0h	RESERVED: Dont Use
10	mpu_rd_dss_l3_bankc	R/W	0h	RESERVED: Dont Use
9	mpu_rd_dss_l3_bankb	R/W	0h	Status of Interrupt from MPU_DSS_L3_BANKB Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
8	mpu_rd_dss_l3_banka	R/W	0h	Status of Interrupt from MPU_DSS_L3_BANKA Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
7	mpu_rd_mss_cr5b_axis	R/W	0h	Status of Interrupt from MPU_MSS_L2_BANKC Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
6	mpu_rd_mss_cr5a_axis	R/W	0h	Status of Interrupt from MPU_MSS_CR5A_AXIS Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
5	mpu_rd_mss_qspi	R/W	0h	Status of Interrupt from MPU_MSS_QSPI Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
4	mpu_rd_mss_pcra	R/W	0h	Status of Interrupt from MPU_MSS_PCRA Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
3	mpu_rd_mss_mbox	R/W	0h	Status of Interrupt from MPU_MSS_MBOX Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
2	mpu_rd_hsm_dthe	R/W	0h	Status of Interrupt from MPU_HSM_DTHE Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
1	mpu_rd_mss_l2_bankb	R/W	0h	Status of Interrupt from MPU_MSS_L2_BANKB Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.
0	mpu_rd_mss_l2_banka	R/W	0h	Status of Interrupt from MPU_MSS_L2_BANKA Set only if Interrupt is unmasked in MSS_PERIPH_ERRAGG_MASK1. Write 0x1 to clear this interrupt.

### 5.2.7.177 MSS\_PERIPH\_ERRAGG\_STATUS\_RAW1 Register (Offset = 66Ch) [Reset = 0000000h]

MSS\_PERIPH\_ERRAGG\_STATUS\_RAW1 is shown in [Table 5-892](#).

Return to the [Summary Table](#).

**Table 5-892. MSS\_PERIPH\_ERRAGG\_STATUS\_RAW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	mpu_rd_hsm	R/W	0h	Raw Status of Interrupt from MPU_HSM. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
15	mpu_rd_dss_mbox	R/W	0h	Raw Status of Interrupt from MPU_DSS_MBOX. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
14	mpu_rd_dss_hwa_proc	R/W	0h	RESERVED:Dont Use
13	mpu_rd_dss_hwa_dma1	R/W	0h	RESERVED:Dont Use
12	mpu_rd_dss_hwa_dma0	R/W	0h	RESERVED:Dont Use
11	mpu_rd_dss_l3_bankd	R/W	0h	RESERVED:Dont Use
10	mpu_rd_dss_l3_bankc	R/W	0h	RESERVED:Dont Use
9	mpu_rd_dss_l3_bankb	R/W	0h	Raw Status of Interrupt from MPU_DSS_L3_BANKB. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
8	mpu_rd_dss_l3_banka	R/W	0h	Raw Status of Interrupt from MPU_DSS_L3_BANKA. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
7	mpu_rd_mss_cr5b_axis	R/W	0h	Raw Status of Interrupt from MPU_MSS_L2_BANKC. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
6	mpu_rd_mss_cr5a_axis	R/W	0h	Raw Status of Interrupt from MPU_MSS_CR5A_AXIS. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
5	mpu_rd_mss_qspi	R/W	0h	Raw Status of Interrupt from MPU_MSS_QSPI. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
4	mpu_rd_mss_pcra	R/W	0h	Raw Status of Interrupt from MPU_MSS_PCRA. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
3	mpu_rd_mss_mbox	R/W	0h	Raw Status of Interrupt from MPU_MSS_MBOX. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
2	mpu_rd_hsm_dthe	R/W	0h	Raw Status of Interrupt from MPU_HSM_DTHE. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
1	mpu_rd_mss_l2_bankb	R/W	0h	Raw Status of Interrupt from MPU_MSS_L2_BANKB. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1
0	mpu_rd_mss_l2_banka	R/W	0h	Raw Status of Interrupt from MPU_MSS_L2_BANKA. Set irrespective if the Interrupt is masked or unmasked in MSS_PERIPH_ERRAGG_MASK1

### 5.2.7.178 MSS\_DMM\_EVENT0\_REG Register (Offset = 670h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT0\_REG is shown in [Table 5-893](#).

Return to the [Summary Table](#).

**Table 5-893. MSS\_DMM\_EVENT0\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel3	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig3	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX1_INT
23-21	RESERVED	R/W	0h	
20	event_sel2	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig2	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX0_INT
15-13	RESERVED	R/W	0h	
12	event_sel1	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig1	R/W	0h	DMM trigger for RSS_CSI2A_SOF_INT1
7-5	RESERVED	R/W	0h	
4	event_sel0	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig0	R/W	0h	DMM trigger for RSS_CSI2A_SOF_INT0

### 5.2.7.179 MSS\_DMM\_EVENT1\_REG Register (Offset = 674h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT1\_REG is shown in [Table 5-894](#).

Return to the [Summary Table](#).

**Table 5-894. MSS\_DMM\_EVENT1\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel7	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig7	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX5_INT
23-21	RESERVED	R/W	0h	
20	event_sel6	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig6	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX4_INT
15-13	RESERVED	R/W	0h	
12	event_sel5	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig5	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX3_INT
7-5	RESERVED	R/W	0h	
4	event_sel4	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig4	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX2_INT

### 5.2.7.180 MSS\_DMM\_EVENT2\_REG Register (Offset = 678h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT2\_REG is shown in [Table 5-895](#).

Return to the [Summary Table](#).

**Table 5-895. MSS\_DMM\_EVENT2\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel11	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig11	R/W	0h	DMM trigger for RSS_ADC_CAPTURE_COMPLETE (to DMA)
23-21	RESERVED	R/W	0h	
20	event_sel10	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig10	R/W	0h	DMM trigger for DFE_FRAME_START_TO_DSS
15-13	RESERVED	R/W	0h	
12	event_sel9	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig9	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX7_INT
7-5	RESERVED	R/W	0h	
4	event_sel8	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig8	R/W	0h	DMM trigger for RSS_CSI2A_EOL_CNTX6_INT

### 5.2.7.181 MSS\_DMM\_EVENT3\_REG Register (Offset = 67Ch) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT3\_REG is shown in [Table 5-896](#).

Return to the [Summary Table](#).

**Table 5-896. MSS\_DMM\_EVENT3\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel15	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig15	R/W	0h	DMM trigger for RSS_ADC_CAPTURE_COMPLETE (to interrupts)
23-21	RESERVED	R/W	0h	
20	event_sel14	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig14	R/W	0h	DMM trigger for FRC_LOGICAL_FRAME_END
15-13	RESERVED	R/W	0h	
12	event_sel13	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig13	R/W	0h	DMM trigger for FRC_LOGICAL_FRAME_START
7-5	RESERVED	R/W	0h	
4	event_sel12	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig12	R/W	0h	DMM trigger for RSS_DATA_CAPTURE_ENABLE_FALL

### 5.2.7.182 MSS\_DMM\_EVENT4\_REG Register (Offset = 680h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT4\_REG is shown in [Table 5-897](#).

Return to the [Summary Table](#).

**Table 5-897. MSS\_DMM\_EVENT4\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel19	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig19	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel18	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig18	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel17	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig17	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel16	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig16	R/W	0h	DMM trigger for ADC_CLK_ENABLE_VALID



### 5.2.7.183 MSS\_DMM\_EVENT5\_REG Register (Offset = 684h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT5\_REG is shown in [Table 5-898](#).

Return to the [Summary Table](#).

**Table 5-898. MSS\_DMM\_EVENT5\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel23	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig23	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel22	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig22	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel21	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig21	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel20	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig20	R/W	0h	DMM trigger Reserved

### 5.2.7.184 MSS\_DMM\_EVENT6\_REG Register (Offset = 688h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT6\_REG is shown in [Table 5-899](#).

Return to the [Summary Table](#).

**Table 5-899. MSS\_DMM\_EVENT6\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel27	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig27	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel26	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig26	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel25	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig25	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel24	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig24	R/W	0h	DMM trigger Reserved

### 5.2.7.185 MSS\_DMM\_EVENT7\_REG Register (Offset = 68Ch) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT7\_REG is shown in [Table 5-900](#).

Return to the [Summary Table](#).

**Table 5-900. MSS\_DMM\_EVENT7\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel31	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig31	R/W	0h	DMM trigger for DSS_HWA_THREAD1_PARAM_DONE
23-21	RESERVED	R/W	0h	
20	event_sel30	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig30	R/W	0h	DMM trigger for DSS_HWA_THREAD1_LOOP
15-13	RESERVED	R/W	0h	
12	event_sel29	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig29	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel28	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig28	R/W	0h	DMM trigger Reserved

### 5.2.7.186 MSS\_DMM\_EVENT8\_REG Register (Offset = 690h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT8\_REG is shown in [Table 5-901](#).

Return to the [Summary Table](#).

**Table 5-901. MSS\_DMM\_EVENT8\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel35	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig35	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel34	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig34	R/W	0h	DMM trigger for DSS_HWA_LOCAL_RAM_ERR
15-13	RESERVED	R/W	0h	
12	event_sel33	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig33	R/W	0h	DMM trigger for DSS_HWA_THREAD2_PARAM_DONE
7-5	RESERVED	R/W	0h	
4	event_sel32	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig32	R/W	0h	DMM trigger for DSS_HWA_THREAD2_LOOP

### 5.2.7.187 MSS\_DMM\_EVENT9\_REG Register (Offset = 694h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT9\_REG is shown in [Table 5-902](#).

Return to the [Summary Table](#).

**Table 5-902. MSS\_DMM\_EVENT9\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel39	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig39	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel38	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig38	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel37	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig37	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel36	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig36	R/W	0h	DMM trigger Reserved

### 5.2.7.188 MSS\_DMM\_EVENT10\_REG Register (Offset = 698h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT10\_REG is shown in [Table 5-903](#).

Return to the [Summary Table](#).

**Table 5-903. MSS\_DMM\_EVENT10\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel43	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig43	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel42	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig42	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel41	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig41	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel40	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig40	R/W	0h	DMM trigger Reserved

### 5.2.7.189 MSS\_DMM\_EVENT11\_REG Register (Offset = 69Ch) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT11\_REG is shown in [Table 5-904](#).

Return to the [Summary Table](#).

**Table 5-904. MSS\_DMM\_EVENT11\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel47	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig47	R/W	0h	DMM trigger For MSS_MCANA_FE_INT source 2
23-21	RESERVED	R/W	0h	
20	event_sel46	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig46	R/W	0h	DMM trigger For MSS_MCANA_FE_INT source 1
15-13	RESERVED	R/W	0h	
12	event_sel45	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig45	R/W	0h	DMM trigger For MSS_MCANA_FE_INT source 0
7-5	RESERVED	R/W	0h	
4	event_sel44	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig44	R/W	0h	DMM trigger Reserved

### 5.2.7.190 MSS\_DMM\_EVENT12\_REG Register (Offset = 6A0h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT12\_REG is shown in [Table 5-905](#).

Return to the [Summary Table](#).

**Table 5-905. MSS\_DMM\_EVENT12\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel51	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig51	R/W	0h	DMM trigger For MSS_MCANA_FE_INT source 1
23-21	RESERVED	R/W	0h	
20	event_sel50	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig50	R/W	0h	DMM trigger For MSS_MCANA_FE_INT source 0
15-13	RESERVED	R/W	0h	
12	event_sel49	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig49	R/W	0h	DMM trigger For MSS_MCANA_INT1
7-5	RESERVED	R/W	0h	
4	event_sel48	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig48	R/W	0h	DMM trigger For MSS_MCANA_INT0



### 5.2.7.191 MSS\_DMM\_EVENT13\_REG Register (Offset = 6A4h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT13\_REG is shown in [Table 5-906](#).

Return to the [Summary Table](#).

**Table 5-906. MSS\_DMM\_EVENT13\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel55	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig55	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel54	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig54	R/W	0h	DMM trigger For MSS_MCANB_INT1
15-13	RESERVED	R/W	0h	
12	event_sel53	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig53	R/W	0h	DMM trigger For MSS_MCANB_INT0
7-5	RESERVED	R/W	0h	
4	event_sel52	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig52	R/W	0h	DMM trigger For MSS_MCANA_FE_INT source 2

### 5.2.7.192 MSS\_DMM\_EVENT14\_REG Register (Offset = 6A8h) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT14\_REG is shown in [Table 5-907](#).

Return to the [Summary Table](#).

**Table 5-907. MSS\_DMM\_EVENT14\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel59	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig59	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel58	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig58	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel57	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig57	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel56	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig56	R/W	0h	DMM trigger Reserved

### 5.2.7.193 MSS\_DMM\_EVENT15\_REG Register (Offset = 6ACh) [Reset = 0XXXXXXXh]

MSS\_DMM\_EVENT15\_REG is shown in [Table 5-908](#).

Return to the [Summary Table](#).

**Table 5-908. MSS\_DMM\_EVENT15\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	event_sel63	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
27-25	RESERVED	R/W	0h	
24	event_trig63	R/W	0h	DMM trigger Reserved
23-21	RESERVED	R/W	0h	
20	event_sel62	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
19-17	RESERVED	R/W	0h	
16	event_trig62	R/W	0h	DMM trigger Reserved
15-13	RESERVED	R/W	0h	
12	event_sel61	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
11-9	RESERVED	R/W	0h	
8	event_trig61	R/W	0h	DMM trigger Reserved
7-5	RESERVED	R/W	0h	
4	event_sel60	R/W	0h	Writing 1'b1 : Selects DMM event_trig as interrupt source. 1'b0 : Selects actual interrupt as interrupt source.
3-1	RESERVED	R/W	0h	
0	event_trig60	R/W	0h	DMM trigger Reserved

### 5.2.7.194 MSS\_TPTC\_BOUNDARY\_CFG Register (Offset = 6B0h) [Reset = 0012X2X2h]

MSS\_TPTC\_BOUNDARY\_CFG is shown in [Table 5-909](#).

Return to the [Summary Table](#).

**Table 5-909. MSS\_TPTC\_BOUNDARY\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R/W	0h	
21-16	tptc_b0_size	R/W	12h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of MSS_TPTC_B0 Example: writing 6'd19 decides boundary to be $2^{19}$ i.e. 512 KB
15-14	RESERVED	R/W	0h	
13-8	tptc_a1_size	R/W	12h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of MSS_TPTC_A1 Example: writing 6'd19 decides boundary to be $2^{19}$ i.e. 512 KB
7-6	RESERVED	R/W	0h	
5-0	tptc_a0_size	R/W	12h	6 bit signal used for deciding the boundary crossing size for CID-RID-SID reordering of MSS_TPTC_A0 Example: writing 6'd19 decides boundary to be $2^{19}$ i.e. 512 KB

### 5.2.7.195 MSS\_TPTC\_XID\_REORDER\_CFG Register (Offset = 6B4h) [Reset = 0000XXXXh]

MSS\_TPTC\_XID\_REORDER\_CFG is shown in [Table 5-910](#).

Return to the [Summary Table](#).

**Table 5-910. MSS\_TPTC\_XID\_REORDER\_CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R/W	0h	
16	tptc_b0_disable	R/W	0h	writing 1'b1 will disable the CID-RID-SID reordering feature for MSS_TPTC_B0
15-9	RESERVED	R/W	0h	
8	tptc_a1_disable	R/W	0h	writing 1'b1 will disable the CID-RID-SID reordering feature for MSS_TPTC_A1
7-1	RESERVED	R/W	0h	
0	tptc_a0_disable	R/W	0h	writing 1'b1 will disable the CID-RID-SID reordering feature for MSS_TPTC_A0

### 5.2.7.196 GPADC\_CTRL Register (Offset = 6B8h) [Reset = 00000FXXh]

GPADC\_CTRL is shown in [Table 5-911](#).

Return to the [Summary Table](#).

**Table 5-911. GPADC\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	
12-8	gpadc_trigin_sel	R/W	Fh	Writing below decimal values to this regiter will select corresponding interrupt as GPADC trigger source. 0: GPIO_0 1: GPIO_1 2: GPIO_2 3: GPIO_3 4: Reserved 5: Reserved 6: Reserved 7: Reserved 8: Reserved 9: Reseved 10: Reserved 11: MSS_RTIA_INT0 12: MSS_RTIA_INT1 13: HW_Sync_FE1 14: Reserved 15: MMR based SW trigger
7-1	RESERVED	R/W	0h	
0	gpadc_sw_trig	R/W	0h	Writing 1'b1 will give MMR based SW trigger to GPADC

### 5.2.7.197 HW\_Sync\_FE\_CTRL Register (Offset = 6BCh) [Reset = 000000XXh]

HW\_Sync\_FE\_CTRL is shown in [Table 5-912](#).

Return to the [Summary Table](#).

**Table 5-912. HW\_Sync\_FE\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	0h	
8	fe2_sel	R/W	0h	Writing 1'b0 : Selects MCANA filter event as HW_Sync_FE2 1'b1 : Selects MCANB filter event as HW_Sync_FE2
7-1	RESERVED	R/W	0h	
0	fe1_sel	R/W	0h	Writing 1'b0 : Selects MCANA filter event as HW_Sync_FE1 1'b1 : Selects MCANB filter event as HW_Sync_FE1

### 5.2.7.198 DEBUGSS\_CSETB\_FLUSH Register (Offset = 6C0h) [Reset = 000000XXh]

DEBUGSS\_CSETB\_FLUSH is shown in [Table 5-913](#).

Return to the [Summary Table](#).

**Table 5-913. DEBUGSS\_CSETB\_FLUSH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10	CSETB_FULL	R	0h	When HIGH indicates that the ETB RAM has overflowed or wrapped around to address zero
9	CSETB_ACQ_COMPLET E	R	0h	When HIGH, indicates that trace acquisition is complete by ETB, that is, the trigger counter is at zero
8	CSETB_FLUSHINACK	R	0h	Return acknowledgement to CSETBFLUSHIN
7-1	RESERVED	R/W	0h	
0	CSETB_FLUSHIN	R/W	0h	External control used to assert the ATB signal AFVALIDS and drain any historical FIFO information on the bus



### 5.2.7.199 ANALOG\_WU\_STATUS\_REG\_POLARITY\_INV Register (Offset = 6C4h) [Reset = 00003D5Ch]

ANALOG\_WU\_STATUS\_REG\_POLARITY\_INV is shown in [Table 5-914](#).

Return to the [Summary Table](#).

**Table 5-914. ANALOG\_WU\_STATUS\_REG\_POLARITY\_INV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	inv_ctrl	R/W	3D5Ch	This register decides the polarity of each status bit before providing to the MSS_ESM. Each bit controls the respective status bit.

### 5.2.7.200 ANALOG\_CLK\_STATUS\_REG\_POLARITY\_INV Register (Offset = 6C8h) [Reset = 00000000h]

ANALOG\_CLK\_STATUS\_REG\_POLARITY\_INV is shown in [Table 5-915](#).

Return to the [Summary Table](#).

**Table 5-915. ANALOG\_CLK\_STATUS\_REG\_POLARITY\_INV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	inv_ctrl	R/W	0h	This register decides the polarity of each status bit before providing to the MSS_ESM. Each bit controls the respective status bit.

### 5.2.7.201 ANALOG\_WU\_STATUS\_REG\_GRP1\_MASK Register (Offset = 6CCh) [Reset = FFFFFFFFh]

ANALOG\_WU\_STATUS\_REG\_GRP1\_MASK is shown in [Table 5-916](#).

Return to the [Summary Table](#).

**Table 5-916. ANALOG\_WU\_STATUS\_REG\_GRP1\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	mask	R/W	FFFFFFFh	Writing 1'b1 : Masks the corresponding status bit before generating a group 1 ESM error. 1'b0 : Unmasks the corresponding status bit before generating a group 1 ESM error.

### 5.2.7.202 ANALOG\_CLK\_STATUS\_REG\_GRP1\_MASK Register (Offset = 6D0h) [Reset = FFFFFFFFh]

ANALOG\_CLK\_STATUS\_REG\_GRP1\_MASK is shown in [Table 5-917](#).

Return to the [Summary Table](#).

**Table 5-917. ANALOG\_CLK\_STATUS\_REG\_GRP1\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	mask	R/W	FFFFFFFh	Writing 1'b1 : Masks the corresponding status bit before generating a group 1 ESM error. 1'b0 : Unmasks the corresponding status bit before generating a group 1 ESM error.

### 5.2.7.203 ANALOG\_WU\_STATUS\_REG\_GRP2\_MASK Register (Offset = 6D4h) [Reset = FFFFFFFFh]

ANALOG\_WU\_STATUS\_REG\_GRP2\_MASK is shown in [Table 5-918](#).

Return to the [Summary Table](#).

**Table 5-918. ANALOG\_WU\_STATUS\_REG\_GRP2\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	mask	R/W	FFFFFFFh	Writing 1'b1 : Masks the corresponding status bit before generating a group 2 ESM error. 1'b0 : Unmasks the corresponding status bit before generating a group 2 ESM error.

### 5.2.7.204 ANALOG\_CLK\_STATUS\_REG\_GRP2\_MASK Register (Offset = 6D8h) [Reset = FFFFFFFFh]

ANALOG\_CLK\_STATUS\_REG\_GRP2\_MASK is shown in [Table 5-919](#).

Return to the [Summary Table](#).

**Table 5-919. ANALOG\_CLK\_STATUS\_REG\_GRP2\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	mask	R/W	FFFFFFFh	Writing 1'b1 : Masks the corresponding status bit before generating a group 2 ESM error. 1'b0 : Unmasks the corresponding status bit before generating a group 2 ESM error.

### 5.2.7.205 NERROR\_MASK Register (Offset = 6DCh) [Reset = 00000000h]

NERROR\_MASK is shown in [Table 5-920](#).

Return to the [Summary Table](#).

**Table 5-920. NERROR\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	mask	R/W	0h	writing 3'b111 will mask the Nerror propagation to pad Writing 3'b000 will unmask the Nerror propagation to pad

### 5.2.7.206 MSS2R5SS\_BUS\_SAFETY\_CTRL Register (Offset = 6E0h) [Reset = 000FXXXh]

MSS2R5SS\_BUS\_SAFETY\_CTRL is shown in [Table 5-921](#).

Return to the [Summary Table](#).

**Table 5-921. MSS2R5SS\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	Fh	Refer to 25xx Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to 25xx Substem Microarch document for more details



### 5.2.7.207 MSS2R5SS\_BUS\_SAFETY\_FI Register (Offset = 6E4h) [Reset = 000000X0h]

MSS2R5SS\_BUS\_SAFETY\_FI is shown in [Table 5-922](#).

Return to the [Summary Table](#).

**Table 5-922. MSS2R5SS\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
23-16	main	R/W	0h	Refer to 25xx Substem Microarch document for more details
15-8	data	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to 25xx Substem Microarch document for more details
4	sec	R/W	0h	Refer to 25xx Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
0	global_main	R/W	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.208 MSS2R5SS\_BUS\_SAFETY\_ERR Register (Offset = 6E8h) [Reset = 00000000h]

MSS2R5SS\_BUS\_SAFETY\_ERR is shown in [Table 5-923](#).

Return to the [Summary Table](#).

**Table 5-923. MSS2R5SS\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to 25xx Substem Microarch document for more details
23-16	sec	R	0h	Refer to 25xx Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.209 MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 6ECh) [Reset = 00000000h]

MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-924](#).

Return to the [Summary Table](#).

**Table 5-924. MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	d0	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.210 MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 6F0h) [Reset = 00000000h]

MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-925](#).

Return to the [Summary Table](#).

**Table 5-925. MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.211 MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 6F4h) [Reset = 00000000h]

MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-926](#).

Return to the [Summary Table](#).

**Table 5-926. MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.212 MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 6F8h) [Reset = 00000000h]

MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-927](#).

Return to the [Summary Table](#).

**Table 5-927. MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.213 MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 6FCh) [Reset = 00000000h]

MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-928](#).

Return to the [Summary Table](#).

**Table 5-928. MSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.214 DSS2R5SS\_BUS\_SAFETY\_CTRL Register (Offset = 700h) [Reset = 000FXXXXh]

DSS2R5SS\_BUS\_SAFETY\_CTRL is shown in [Table 5-929](#).

Return to the [Summary Table](#).

**Table 5-929. DSS2R5SS\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	Fh	Refer to 25xx Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to 25xx Substem Microarch document for more details



### 5.2.7.215 DSS2R5SS\_BUS\_SAFETY\_FI Register (Offset = 704h) [Reset = 000000X0h]

DSS2R5SS\_BUS\_SAFETY\_FI is shown in [Table 5-930](#).

Return to the [Summary Table](#).

**Table 5-930. DSS2R5SS\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
23-16	main	R/W	0h	Refer to 25xx Substem Microarch document for more details
15-8	data	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to 25xx Substem Microarch document for more details
4	sec	R/W	0h	Refer to 25xx Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
0	global_main	R/W	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.216 DSS2R5SS\_BUS\_SAFETY\_ERR Register (Offset = 708h) [Reset = 00000000h]

DSS2R5SS\_BUS\_SAFETY\_ERR is shown in [Table 5-931](#).

Return to the [Summary Table](#).

**Table 5-931. DSS2R5SS\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to 25xx Substem Microarch document for more details
23-16	sec	R	0h	Refer to 25xx Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.217 DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 70Ch) [Reset = 00000000h]

DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-932](#).

Return to the [Summary Table](#).

**Table 5-932. DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	d0	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.218 DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 710h) [Reset = 00000000h]

DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-933](#).

Return to the [Summary Table](#).

**Table 5-933. DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.219 DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 714h) [Reset = 00000000h]

DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-934](#).

Return to the [Summary Table](#).

**Table 5-934. DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.220 DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 718h) [Reset = 00000000h]

DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-935](#).

Return to the [Summary Table](#).

**Table 5-935. DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.221 DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 71Ch) [Reset = 00000000h]

DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-936](#).

Return to the [Summary Table](#).

**Table 5-936. DSS2R5SS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.222 MSS\_DMM\_ACCESS\_MODE Register (Offset = 720h) [Reset = 0000001Xh]

MSS\_DMM\_ACCESS\_MODE is shown in [Table 5-937](#).

Return to the [Summary Table](#).

**Table 5-937. MSS\_DMM\_ACCESS\_MODE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	dmmb_sel	R/W	1h	writing 1'b0 : ensures all the accesses from DMMB are user-mode writing 1'b1 : ensures all the accesses from DMMB are privilege mode
3-1	RESERVED	R/W	0h	
0	dmma_sel	R/W	1h	writing 1'b0 : ensures all the accesses from DMMA are user-mode writing 1'b1 : ensures all the accesses from DMMA are privilege mode



### 5.2.7.223 CPSW\_HW\_TRIG\_CTRL Register (Offset = 724h) [Reset = 00000000h]

CPSW\_HW\_TRIG\_CTRL is shown in [Table 5-938](#).

Return to the [Summary Table](#).

**Table 5-938. CPSW\_HW\_TRIG\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	CPSW_HW_TRIG_CTRL	R/W	0h	Bits [1:0] are used for Genf select line bits 2'b00 : Genf0 2'b01 : Genf1 2'b10 : Genf2 2'b11 : Not used Bit2 is used to select Genf or CPSW_HW_TRIG_VAL

### 5.2.7.224 CPSW\_HW\_TRIG\_VAL Register (Offset = 728h) [Reset = 00000000h]

CPSW\_HW\_TRIG\_VAL is shown in [Table 5-939](#).

Return to the [Summary Table](#).

**Table 5-939. CPSW\_HW\_TRIG\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	CPSW_HW_TRIG_VAL	R/W	0h	Trigger

### 5.2.7.225 CPSW\_TRIG\_CAPTURE\_COUNT Register (Offset = 72Ch) [Reset = 00000005h]

CPSW\_TRIG\_CAPTURE\_COUNT is shown in [Table 5-940](#).

Return to the [Summary Table](#).

**Table 5-940. CPSW\_TRIG\_CAPTURE\_COUNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	trig_count	R/W	5h	Configuration of number of cycles for pulse extender of TRIG signal to CPSW

### 5.2.7.226 MSS\_L2\_C\_BUS\_SAFETY\_CTRL Register (Offset = 730h) [Reset = 001FXXXXh]

MSS\_L2\_C\_BUS\_SAFETY\_CTRL is shown in [Table 5-941](#).

Return to the [Summary Table](#).

**Table 5-941. MSS\_L2\_C\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	1Fh	Refer to 25xx Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.227 MSS\_L2\_C\_BUS\_SAFETY\_FI Register (Offset = 734h) [Reset = 000000X0h]

MSS\_L2\_C\_BUS\_SAFETY\_FI is shown in [Table 5-942](#).

Return to the [Summary Table](#).

**Table 5-942. MSS\_L2\_C\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
23-16	main	R/W	0h	Refer to 25xx Substem Microarch document for more details
15-8	data	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to 25xx Substem Microarch document for more details
4	sec	R/W	0h	Refer to 25xx Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
0	global_main	R/W	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.228 MSS\_L2\_C\_BUS\_SAFETY\_ERR Register (Offset = 738h) [Reset = 00000000h]

MSS\_L2\_C\_BUS\_SAFETY\_ERR is shown in [Table 5-943](#).

Return to the [Summary Table](#).

**Table 5-943. MSS\_L2\_C\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to 25xx Substem Microarch document for more details
23-16	sec	R	0h	Refer to 25xx Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.229 MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 73Ch) [Reset = 00000000h]

MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-944](#).

Return to the [Summary Table](#).

**Table 5-944. MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	d0	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.230 MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 740h) [Reset = 00000000h]

MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-945](#).

Return to the [Summary Table](#).

**Table 5-945. MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details



### 5.2.7.231 MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 744h) [Reset = 00000000h]

MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-946](#).

Return to the [Summary Table](#).

**Table 5-946. MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.232 MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 748h) [Reset = 00000000h]

MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-947](#).

Return to the [Summary Table](#).

**Table 5-947. MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.233 MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 74Ch) [Reset = 00000000h]

MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-948](#).

Return to the [Summary Table](#).

**Table 5-948. MSS\_L2\_C\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.234 R5SS2DSS\_BUS\_SAFETY\_CTRL Register (Offset = 750h) [Reset = 000FXXXXh]

R5SS2DSS\_BUS\_SAFETY\_CTRL is shown in [Table 5-949](#).

Return to the [Summary Table](#).

**Table 5-949. R5SS2DSS\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	Fh	Refer to 25xx Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.235 R5SS2DSS\_BUS\_SAFETY\_FI Register (Offset = 754h) [Reset = 000000X0h]

R5SS2DSS\_BUS\_SAFETY\_FI is shown in [Table 5-950](#).

Return to the [Summary Table](#).

**Table 5-950. R5SS2DSS\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
23-16	main	R/W	0h	Refer to 25xx Substem Microarch document for more details
15-8	data	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to 25xx Substem Microarch document for more details
4	sec	R/W	0h	Refer to 25xx Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
0	global_main	R/W	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.236 R5SS2DSS\_BUS\_SAFETY\_ERR Register (Offset = 758h) [Reset = 00000000h]

R5SS2DSS\_BUS\_SAFETY\_ERR is shown in [Table 5-951](#).

Return to the [Summary Table](#).

**Table 5-951. R5SS2DSS\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to 25xx Substem Microarch document for more details
23-16	sec	R	0h	Refer to 25xx Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.237 R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 75Ch) [Reset = 00000000h]

R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-952](#).

Return to the [Summary Table](#).

**Table 5-952. R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	d0	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.238 R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 760h) [Reset = 00000000h]

R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-953](#).

Return to the [Summary Table](#).

**Table 5-953. R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details



### 5.2.7.239 R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 764h) [Reset = 00000000h]

R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-954](#).

Return to the [Summary Table](#).

**Table 5-954. R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.240 R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 768h) [Reset = 00000000h]

R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-955](#).

Return to the [Summary Table](#).

**Table 5-955. R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.241 R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 76Ch) [Reset = 00000000h]

R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-956](#).

Return to the [Summary Table](#).

**Table 5-956. R5SS2DSS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.242 R5SS2MSS\_BUS\_SAFETY\_CTRL Register (Offset = 770h) [Reset = 000FXXXh]

R5SS2MSS\_BUS\_SAFETY\_CTRL is shown in [Table 5-957](#).

Return to the [Summary Table](#).

**Table 5-957. R5SS2MSS\_BUS\_SAFETY\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R/W	0h	
23-16	type	R	Fh	Refer to 25xx Substem Microarch document for more details
15-9	RESERVED	R/W	0h	
8	err_clear	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-3	RESERVED	R/W	0h	
2-0	enable	R/W	7h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.243 R5SS2MSS\_BUS\_SAFETY\_FI Register (Offset = 774h) [Reset = 000000X0h]

R5SS2MSS\_BUS\_SAFETY\_FI is shown in [Table 5-958](#).

Return to the [Summary Table](#).

**Table 5-958. R5SS2MSS\_BUS\_SAFETY\_FI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
23-16	main	R/W	0h	Refer to 25xx Substem Microarch document for more details
15-8	data	R/W	0h	Refer to 25xx Substem Microarch document for more details
7-6	RESERVED	R/W	0h	
5	ded	R/W	0h	Refer to 25xx Substem Microarch document for more details
4	sec	R/W	0h	Refer to 25xx Substem Microarch document for more details
3	global_safe_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
2	global_main_req	R/W	0h	Refer to 25xx Substem Microarch document for more details
1	global_safe	R/W	0h	Refer to 25xx Substem Microarch document for more details
0	global_main	R/W	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.244 R5SS2MSS\_BUS\_SAFETY\_ERR Register (Offset = 778h) [Reset = 00000000h]

R5SS2MSS\_BUS\_SAFETY\_ERR is shown in [Table 5-959](#).

Return to the [Summary Table](#).

**Table 5-959. R5SS2MSS\_BUS\_SAFETY\_ERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	ded	R	0h	Refer to 25xx Substem Microarch document for more details
23-16	sec	R	0h	Refer to 25xx Substem Microarch document for more details
15-8	comp_check	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	comp_err	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.245 R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register (Offset = 77Ch) [Reset = 00000000h]

R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 is shown in [Table 5-960](#).

Return to the [Summary Table](#).

**Table 5-960. R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_DATA0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	d1	R	0h	Refer to 25xx Substem Microarch document for more details
7-0	d0	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.246 R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register (Offset = 780h) [Reset = 00000000h]

R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_CMD is shown in [Table 5-961](#).

Return to the [Summary Table](#).

**Table 5-961. R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_CMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details



### 5.2.7.247 R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register (Offset = 784h) [Reset = 00000000h]

R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_WRITE is shown in [Table 5-962](#).

Return to the [Summary Table](#).

**Table 5-962. R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_WRITE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.248 R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_READ Register (Offset = 788h) [Reset = 00000000h]

R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_READ is shown in [Table 5-963](#).

Return to the [Summary Table](#).

**Table 5-963. R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_READ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.249 R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register (Offset = 78Ch) [Reset = 00000000h]

R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP is shown in [Table 5-964](#).

Return to the [Summary Table](#).

**Table 5-964. R5SS2MSS\_BUS\_SAFETY\_ERR\_STAT\_WRITERESP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Refer to 25xx Substem Microarch document for more details

### 5.2.7.250 NW\_PACKET\_COUNT Register (Offset = 790h) [Reset = 00000000h]

NW\_PACKET\_COUNT is shown in [Table 5-965](#).

Return to the [Summary Table](#).

**Table 5-965. NW\_PACKET\_COUNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	stat	R	0h	Gives the count of the ping-pong switch events of the network packet buffer

### 5.2.7.251 NW\_PACKET\_COUNT\_RESET Register (Offset = 794h) [Reset = 00000000h]

NW\_PACKET\_COUNT\_RESET is shown in [Table 5-966](#).

Return to the [Summary Table](#).

**Table 5-966. NW\_PACKET\_COUNT\_RESET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	undefined	R/W	0h	Software reset for Counter maintaining Ping pong switch events

### 5.2.7.252 CPSW\_CRC\_PING\_ADDR Register (Offset = 798h) [Reset = 00000000h]

CPSW\_CRC\_PING\_ADDR is shown in [Table 5-967](#).

Return to the [Summary Table](#).

**Table 5-967. CPSW\_CRC\_PING\_ADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	undefined	R/W	0h	This denotes the last address of the data Read. This depends on the Packet size . Example : If packet size is 3072 bytes or 3KB and CRC is disabled , this needs to be programmed as 3064 or hBF8 If packet size is 3072 bytes or 3KB and 16 bit CRC , this needs to be programmed as hC00 Address must be 4 byte aligned for 32 bit CRC Address must be 2 byte aligned for 16 bit CRC

### 5.2.7.253 CPSW\_CRC\_PONG\_ADDR Register (Offset = 79Ch) [Reset = 00000000h]

CPSW\_CRC\_PONG\_ADDR is shown in [Table 5-968](#).

Return to the [Summary Table](#).

**Table 5-968. CPSW\_CRC\_PONG\_ADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	
15-0	undefined	R/W	0h	This denotes the last address of the data Read. This depends on the Packet size . Example : If packet size is 3072 bytes or 3KB and CRC is disabled , this needs to programmed as 3064 or hBF8 If packet size is 3072 bytes or 3KB and 16 bit CRC , this needs to be programmed as hC00 Address must be 4 byte aligned for 32 bit CRC Address must be 2 byte aligned for 16 bit CRC

### 5.2.7.254 R5\_CONTROL Register (Offset = 800h) [Reset = 00XXXXXXh]

R5\_CONTROL is shown in [Table 5-969](#).

Return to the [Summary Table](#).

**Table 5-969. R5\_CONTROL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R/W	0h	
26-24	rom_wait_state	R/W	0h	writing '111' enables a single cycle wait state with respect to CR5A_clk for rom access. This needs to be set when R5 clock is at 400MHZ and Interconnect-clk is at 200MHZ. (because it is a timing issue in this scenario)
23-19	RESERVED	R/W	0h	
18-16	reset_fsm_trigger	R/W	0h	Write pulse bit field: writing 3'b111 will trigger the reset FSM. Reset FSM ensures reset to R5SS and inturn ensures the latching of lock_step and also mem_swap bit
15-11	RESERVED	R/W	0h	
10-8	lock_step_switch_wait	R/W	0h	writing 3'b111 ensures switch happens only after R5SS reset. Orelse it will be a immediate switch.
7-3	RESERVED	R/W	0h	
2-0	lock_step	R/W	7h	writing 3'b000 ensures R5 to be in Dual-Core mode. Note: The change happens after the R5SS reset assertion if R5_CONTROL_lock_step_switch_wait is set. Or else the switching to Dual-core happens on the fly.



### 5.2.7.255 R5\_ROM\_ECLIPSE Register (Offset = 804h) [Reset = 000000XXh]

R5\_ROM\_ECLIPSE is shown in [Table 5-970](#).

Return to the [Summary Table](#).

**Table 5-970. R5\_ROM\_ECLIPSE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R/W	0h	
10-8	memswap_wait	R/W	0h	writing 3'b111 ensures ROM-Eclipsing happens only after R5SS reset. Orelse it will be a immediate change.
7-3	RESERVED	R/W	0h	
2-0	memswap	R/W	0h	writing '111' ensures eclipsing of CR5A_ROM immediately if memswap_wait is not set. If memswap_wait is set then ROM is eclipsed after R5SS reset assertion.

### 5.2.7.256 R5\_COREA\_HALT Register (Offset = 808h) [Reset = 00000007h]

R5\_COREA\_HALT is shown in [Table 5-971](#).

Return to the [Summary Table](#).

**Table 5-971. R5\_COREA\_HALT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	halt	R/W	7h	writing '000' will un halt CR5A. This register should be written only once.

### 5.2.7.257 R5\_COREB\_HALT Register (Offset = 80Ch) [Reset = 00000007h]

R5\_COREB\_HALT is shown in [Table 5-972](#).

Return to the [Summary Table](#).

**Table 5-972. R5\_COREB\_HALT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	halt	R/W	7h	RESERVED: Dont Use

### 5.2.7.258 R5\_STATUS\_REG Register (Offset = 810h) [Reset = 000001XXh]

R5\_STATUS\_REG is shown in [Table 5-973](#).

Return to the [Summary Table](#).

**Table 5-973. R5\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	lock_step	R	1h	Reading 1: confirms R5SS is in lockstep mode. Reading 0: confirms R5SS is in Dual-core mode.
7-1	RESERVED	R	0h	
0	memswap	R	0h	reading 1: confirms ROM is Eclipsed from with RAM for R5.

### 5.2.7.259 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-974](#).

Return to the [Summary Table](#).

**Table 5-974. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	[2:0] writing 3'b111 gates the clock to redundant-safe bridges and inteconnects [6:4] RESERVED [8]: Mux Select to RSS_Interrupt_Map to choose between Pbist_done and MSS_RTIC Interrupt [9]:writing 1'b1 clock-gates HSM hard-macro [31:7] Reserved for HW R&D

### 5.2.7.260 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-975](#).

Return to the [Summary Table](#).

**Table 5-975. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.7.261 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-976](#).

Return to the [Summary Table](#).

**Table 5-976. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D

### 5.2.7.262 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-977](#).

Return to the [Summary Table](#).

**Table 5-977. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D



### 5.2.7.263 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-978](#).

Return to the [Summary Table](#).

**Table 5-978. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.7.264 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-979](#).

Return to the [Summary Table](#).

**Table 5-979. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

### 5.2.7.265 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-980](#).

Return to the [Summary Table](#).

**Table 5-980. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

### 5.2.7.266 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-981](#).

Return to the [Summary Table](#).

**Table 5-981. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.7.267 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-982](#).

Return to the [Summary Table](#).

**Table 5-982. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	proc	R/W	0h	Write pulse bit field: For bits 0 to 7: Wrting 1'b1 : Generates pulse interrupt to corresponding proc from MSS_CR5A. For bits 8 to 15: RESERVED

### 5.2.7.268 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-983](#).

Return to the [Summary Table](#).

**Table 5-983. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.7.269 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-984](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-984. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

### 5.2.7.270 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-985](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-985. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component



### 5.2.7.271 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-986](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-986. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.7.272 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-987](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-987. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.7.273 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-988](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-988. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.7.274 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-989](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-989. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.7.275 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-990](#).

Return to the [Summary Table](#).

EOI register

**Table 5-990. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.7.276 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-991](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-991. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.7.277 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-992](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-992. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault

### 5.2.7.278 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-993](#).

[Return to the Summary Table.](#)

Fault Attribute Status register

**Table 5-993. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.



### 5.2.7.279 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-994](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-994. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

## 5.2.8 MSS\_IOMUX Registers

Table 5-995 lists the memory-mapped registers for the MSS\_IOMUX registers. All register offset addresses not listed in Table 5-995 should be considered as reserved locations and the register contents should not be modified.

**Table 5-995. MSS\_IOMUX Registers**

Offset	Acronym	Register Name	Section
0h	PADAA_cfg_reg		<a href="#">Section 5.2.8.1</a>
4h	PADAB_cfg_reg		<a href="#">Section 5.2.8.2</a>
8h	PADAC_cfg_reg		<a href="#">Section 5.2.8.3</a>
Ch	PADAD_cfg_reg		<a href="#">Section 5.2.8.4</a>
10h	PADAE_cfg_reg		<a href="#">Section 5.2.8.5</a>
14h	PADAF_cfg_reg		<a href="#">Section 5.2.8.6</a>
18h	PADAG_cfg_reg		<a href="#">Section 5.2.8.7</a>
1Ch	PADAH_cfg_reg		<a href="#">Section 5.2.8.8</a>
20h	PADAI_cfg_reg		<a href="#">Section 5.2.8.9</a>
24h	PADAJ_cfg_reg		<a href="#">Section 5.2.8.10</a>
28h	PADAK_cfg_reg		<a href="#">Section 5.2.8.11</a>
2Ch	PADAL_cfg_reg		<a href="#">Section 5.2.8.12</a>
30h	PADAM_cfg_reg		<a href="#">Section 5.2.8.13</a>
34h	PADAN_cfg_reg		<a href="#">Section 5.2.8.14</a>
38h	PADAO_cfg_reg		<a href="#">Section 5.2.8.15</a>
3Ch	PADAP_cfg_reg		<a href="#">Section 5.2.8.16</a>
40h	PADAQ_cfg_reg		<a href="#">Section 5.2.8.17</a>
44h	PADAR_cfg_reg	OPen Drain Pad	<a href="#">Section 5.2.8.18</a>
48h	PADAS_cfg_reg	OPen Drain Pad	<a href="#">Section 5.2.8.19</a>
4Ch	PADAT_cfg_reg	OPen Drain Pad	<a href="#">Section 5.2.8.20</a>
50h	PDAU_cfg_reg		<a href="#">Section 5.2.8.21</a>
54h	PDAV_cfg_reg		<a href="#">Section 5.2.8.22</a>
58h	PDAW_cfg_reg		<a href="#">Section 5.2.8.23</a>
5Ch	PADAX_cfg_reg		<a href="#">Section 5.2.8.24</a>
60h	PADAY_cfg_reg		<a href="#">Section 5.2.8.25</a>
64h	PADAZ_cfg_reg		<a href="#">Section 5.2.8.26</a>
68h	PADBA_cfg_reg		<a href="#">Section 5.2.8.27</a>
6Ch	PADBB_cfg_reg		<a href="#">Section 5.2.8.28</a>
70h	PADBC_cfg_reg		<a href="#">Section 5.2.8.29</a>
74h	PABD_cfg_reg		<a href="#">Section 5.2.8.30</a>
78h	PADBE_cfg_reg		<a href="#">Section 5.2.8.31</a>
7Ch	PADBF_cfg_reg		<a href="#">Section 5.2.8.32</a>
80h	PADBG_cfg_reg		<a href="#">Section 5.2.8.33</a>
84h	PADBH_cfg_reg		<a href="#">Section 5.2.8.34</a>
88h	PADBI_cfg_reg		<a href="#">Section 5.2.8.35</a>
8Ch	PADBJ_cfg_reg		<a href="#">Section 5.2.8.36</a>
90h	PADBK_cfg_reg		<a href="#">Section 5.2.8.37</a>
94h	PADBL_cfg_reg		<a href="#">Section 5.2.8.38</a>
98h	PADBM_cfg_reg		<a href="#">Section 5.2.8.39</a>
9Ch	PADBN_cfg_reg		<a href="#">Section 5.2.8.40</a>

**Table 5-995. MSS\_IOMUX Registers (continued)**

Offset	Acronym	Register Name	Section
A0h	PADBO_cfg_reg		<a href="#">Section 5.2.8.41</a>
A4h	PADBP_cfg_reg		<a href="#">Section 5.2.8.42</a>
A8h	PADBQ_cfg_reg		<a href="#">Section 5.2.8.43</a>
ACh	PADBR_cfg_reg		<a href="#">Section 5.2.8.44</a>
B0h	PADBS_cfg_reg		<a href="#">Section 5.2.8.45</a>
B4h	PADBT_cfg_reg		<a href="#">Section 5.2.8.46</a>
B8h	PADBU_cfg_reg		<a href="#">Section 5.2.8.47</a>
BCh	PADBV_cfg_reg		<a href="#">Section 5.2.8.48</a>
C0h	PADBW_cfg_reg		<a href="#">Section 5.2.8.49</a>
C4h	PADBX_cfg_reg		<a href="#">Section 5.2.8.50</a>
C8h	PADBY_cfg_reg		<a href="#">Section 5.2.8.51</a>
CCh	PADBZ_cfg_reg		<a href="#">Section 5.2.8.52</a>
D0h	PADCA_cfg_reg		<a href="#">Section 5.2.8.53</a>
D4h	PADCB_cfg_reg		<a href="#">Section 5.2.8.54</a>
D8h	PADCC_cfg_reg		<a href="#">Section 5.2.8.55</a>
DCh	PADCD_cfg_reg		<a href="#">Section 5.2.8.56</a>
E0h	PADCE_cfg_reg		<a href="#">Section 5.2.8.57</a>
E4h	PADCF_cfg_reg		<a href="#">Section 5.2.8.58</a>
E8h	PADCG_cfg_reg		<a href="#">Section 5.2.8.59</a>
ECh	PADCH_cfg_reg		<a href="#">Section 5.2.8.60</a>
F0h	PADCI_cfg_reg		<a href="#">Section 5.2.8.61</a>
F4h	PADCJ_cfg_reg		<a href="#">Section 5.2.8.62</a>
F8h	PADCK_cfg_reg		<a href="#">Section 5.2.8.63</a>
FCh	PADCL_cfg_reg		<a href="#">Section 5.2.8.64</a>
100h	PADCM_cfg_reg		<a href="#">Section 5.2.8.65</a>
104h	PADCN_cfg_reg		<a href="#">Section 5.2.8.66</a>
108h	PADCO_cfg_reg		<a href="#">Section 5.2.8.67</a>
10Ch	PADCP_cfg_reg		<a href="#">Section 5.2.8.68</a>
110h	PADCQ_cfg_reg		<a href="#">Section 5.2.8.69</a>
114h	PADCR_cfg_reg		<a href="#">Section 5.2.8.70</a>
118h	PADCS_cfg_reg		<a href="#">Section 5.2.8.71</a>
11Ch	PADCT_cfg_reg		<a href="#">Section 5.2.8.72</a>
120h	PADCU_cfg_reg		<a href="#">Section 5.2.8.73</a>
124h	PADCV_cfg_reg		<a href="#">Section 5.2.8.74</a>
128h	PADCW_cfg_reg		<a href="#">Section 5.2.8.75</a>
12Ch	PADCX_cfg_reg		<a href="#">Section 5.2.8.76</a>
130h	PADCY_cfg_reg		<a href="#">Section 5.2.8.77</a>
134h	PADCZ_cfg_reg		<a href="#">Section 5.2.8.78</a>
138h	PADDA_cfg_reg		<a href="#">Section 5.2.8.79</a>
13Ch	PADDB_cfg_reg		<a href="#">Section 5.2.8.80</a>
140h	PADDC_cfg_reg		<a href="#">Section 5.2.8.81</a>
144h	PADDD_cfg_reg		<a href="#">Section 5.2.8.82</a>
148h	PADDE_cfg_reg		<a href="#">Section 5.2.8.83</a>
14Ch	PADDF_cfg_reg		<a href="#">Section 5.2.8.84</a>
150h	PADDG_cfg_reg		<a href="#">Section 5.2.8.85</a>

**Table 5-995. MSS\_IOMUX Registers (continued)**

Offset	Acronym	Register Name	Section
154h	PADDH_cfg_reg		<a href="#">Section 5.2.8.86</a>
158h	PADDI_cfg_reg		<a href="#">Section 5.2.8.87</a>
15Ch	PADDJ_cfg_reg		<a href="#">Section 5.2.8.88</a>
160h	PADDK_cfg_reg		<a href="#">Section 5.2.8.89</a>
164h	PADDL_cfg_reg		<a href="#">Section 5.2.8.90</a>
168h	PADDM_cfg_reg		<a href="#">Section 5.2.8.91</a>
16Ch	PADDN_cfg_reg		<a href="#">Section 5.2.8.92</a>
170h	PADDO_cfg_reg		<a href="#">Section 5.2.8.93</a>
174h	PADDP_cfg_reg		<a href="#">Section 5.2.8.94</a>
178h	PADDQ_cfg_reg		<a href="#">Section 5.2.8.95</a>
17Ch	PADDR_cfg_reg		<a href="#">Section 5.2.8.96</a>
180h	PADDS_cfg_reg		<a href="#">Section 5.2.8.97</a>
184h	PADDT_cfg_reg		<a href="#">Section 5.2.8.98</a>
188h	PADDU_cfg_reg		<a href="#">Section 5.2.8.99</a>
18Ch	PADDV_cfg_reg		<a href="#">Section 5.2.8.100</a>
190h	PADDW_cfg_reg		<a href="#">Section 5.2.8.101</a>
194h	PADDX_cfg_reg		<a href="#">Section 5.2.8.102</a>
198h	PADDY_cfg_reg		<a href="#">Section 5.2.8.103</a>
19Ch	PADDZ_cfg_reg		<a href="#">Section 5.2.8.104</a>
1A0h	PADEA_cfg_reg		<a href="#">Section 5.2.8.105</a>
1A4h	PADEB_cfg_reg		<a href="#">Section 5.2.8.106</a>
1A8h	PADEC_cfg_reg		<a href="#">Section 5.2.8.107</a>
1ACh	PADED_cfg_reg		<a href="#">Section 5.2.8.108</a>
1B0h	PADEE_cfg_reg		<a href="#">Section 5.2.8.109</a>
1B4h	PADEF_cfg_reg		<a href="#">Section 5.2.8.110</a>
1B8h	PADEG_cfg_reg		<a href="#">Section 5.2.8.111</a>
1BCh	PADEH_cfg_reg		<a href="#">Section 5.2.8.112</a>
1C0h	PADEI_cfg_reg		<a href="#">Section 5.2.8.113</a>
1C4h	PADEJ_cfg_reg		<a href="#">Section 5.2.8.114</a>
1C8h	PADEK_cfg_reg		<a href="#">Section 5.2.8.115</a>
1CCh	PADEL_cfg_reg		<a href="#">Section 5.2.8.116</a>
1D0h	PADEM_cfg_reg		<a href="#">Section 5.2.8.117</a>
1D4h	PADEN_cfg_reg		<a href="#">Section 5.2.8.118</a>
1D8h	PADEO_cfg_reg		<a href="#">Section 5.2.8.119</a>
1DCh	PADEP_cfg_reg		<a href="#">Section 5.2.8.120</a>
1F0h	USERMODEEN		<a href="#">Section 5.2.8.121</a>
1F4h	PADGLBLCFGREG		<a href="#">Section 5.2.8.122</a>
1F8h	IOCFGKICK0		<a href="#">Section 5.2.8.123</a>
1FCh	IOCFGKICK1		<a href="#">Section 5.2.8.124</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-996](#) shows the codes that are used for access types in this section.

**Table 5-996. MSS\_IOMUX Access Type Codes**

Access Type	Code	Description
Read Type		

**Table 5-996. MSS\_IOMUX Access Type Codes  
(continued)**

Access Type	Code	Description
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.8.1 PADAA\_cfg\_reg Register (Offset = 0h) [Reset = 00000000h]

PADAA\_cfg\_reg is shown in [Table 5-997](#).

Return to the [Summary Table](#).

**Table 5-997. PADAA\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.2 PADAB\_cfg\_reg Register (Offset = 4h) [Reset = 00000000h]

PADAB\_cfg\_reg is shown in [Table 5-998](#).

Return to the [Summary Table](#).

**Table 5-998. PADAB\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.3 PADAC\_cfg\_reg Register (Offset = 8h) [Reset = 00000000h]

PADAC\_cfg\_reg is shown in [Table 5-999](#).

Return to the [Summary Table](#).

**Table 5-999. PADAC\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.4 PADAD\_cfg\_reg Register (Offset = Ch) [Reset = 00000000h]

PADAD\_cfg\_reg is shown in [Table 5-1000](#).

Return to the [Summary Table](#).

**Table 5-1000. PADAD\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.5 PADAE\_cfg\_reg Register (Offset = 10h) [Reset = 00000000h]

PADAE\_cfg\_reg is shown in [Table 5-1001](#).

Return to the [Summary Table](#).

**Table 5-1001. PADAE\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.6 PADAF\_cfg\_reg Register (Offset = 14h) [Reset = 00000000h]

PADAF\_cfg\_reg is shown in [Table 5-1002](#).

Return to the [Summary Table](#).

**Table 5-1002. PADAF\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.7 PADAG\_cfg\_reg Register (Offset = 18h) [Reset = 00000000h]

PADAG\_cfg\_reg is shown in [Table 5-1003](#).

Return to the [Summary Table](#).

**Table 5-1003. PADAG\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.8 PADAH\_cfg\_reg Register (Offset = 1Ch) [Reset = 00000000h]

PADAH\_cfg\_reg is shown in [Table 5-1004](#).

Return to the [Summary Table](#).

**Table 5-1004. PADAH\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.9 PADAI\_cfg\_reg Register (Offset = 20h) [Reset = 00000000h]

PADAI\_cfg\_reg is shown in [Table 5-1005](#).

Return to the [Summary Table](#).

**Table 5-1005. PADAI\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.10 PADAJ\_cfg\_reg Register (Offset = 24h) [Reset = 00000000h]

PADAJ\_cfg\_reg is shown in [Table 5-1006](#).

Return to the [Summary Table](#).

**Table 5-1006. PADAJ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.11 PADAK\_cfg\_reg Register (Offset = 28h) [Reset = 00000000h]

PADAK\_cfg\_reg is shown in [Table 5-1007](#).

Return to the [Summary Table](#).

**Table 5-1007. PADAK\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.12 PADAL\_cfg\_reg Register (Offset = 2Ch) [Reset = 00000000h]

PADAL\_cfg\_reg is shown in [Table 5-1008](#).

Return to the [Summary Table](#).

**Table 5-1008. PADAL\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.13 PADAM\_cfg\_reg Register (Offset = 30h) [Reset = 00000000h]

PADAM\_cfg\_reg is shown in [Table 5-1009](#).

Return to the [Summary Table](#).

**Table 5-1009. PADAM\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

#### 5.2.8.14 PADAN\_cfg\_reg Register (Offset = 34h) [Reset = 00000000h]

PADAN\_cfg\_reg is shown in [Table 5-1010](#).

Return to the [Summary Table](#).

**Table 5-1010. PADAN\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.15 PADO\_cfg\_reg Register (Offset = 38h) [Reset = 00000000h]

PADO\_cfg\_reg is shown in [Table 5-1011](#).

Return to the [Summary Table](#).

**Table 5-1011. PADO\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.16 PADAP\_cfg\_reg Register (Offset = 3Ch) [Reset = 00000000h]

PADAP\_cfg\_reg is shown in [Table 5-1012](#).

Return to the [Summary Table](#).

**Table 5-1012. PADAP\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.17 PADAQ\_cfg\_reg Register (Offset = 40h) [Reset = 00000000h]

PADAQ\_cfg\_reg is shown in [Table 5-1013](#).

Return to the [Summary Table](#).

**Table 5-1013. PADAQ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.18 PADAR\_cfg\_reg Register (Offset = 44h) [Reset = 00000000h]

PADAR\_cfg\_reg is shown in [Table 5-1014](#).

Return to the [Summary Table](#).

**Table 5-1014. PADAR\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.19 PADAS\_cfg\_reg Register (Offset = 48h) [Reset = 00000000h]

PADAS\_cfg\_reg is shown in [Table 5-1015](#).

Return to the [Summary Table](#).

**Table 5-1015. PADAS\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.20 PADAT\_cfg\_reg Register (Offset = 4Ch) [Reset = 00000000h]

PADAT\_cfg\_reg is shown in [Table 5-1016](#).

Return to the [Summary Table](#).

**Table 5-1016. PADAT\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.21 PADAU\_cfg\_reg Register (Offset = 50h) [Reset = 00000000h]

PADAU\_cfg\_reg is shown in [Table 5-1017](#).

Return to the [Summary Table](#).

**Table 5-1017. PADAU\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.22 PADAV\_cfg\_reg Register (Offset = 54h) [Reset = 00000000h]

PADAV\_cfg\_reg is shown in [Table 5-1018](#).

Return to the [Summary Table](#).

**Table 5-1018. PADAV\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.23 PADAW\_cfg\_reg Register (Offset = 58h) [Reset = 00000000h]

PADAW\_cfg\_reg is shown in [Table 5-1019](#).

Return to the [Summary Table](#).

**Table 5-1019. PADAW\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.24 PADAX\_cfg\_reg Register (Offset = 5Ch) [Reset = 00000000h]

PADAX\_cfg\_reg is shown in [Table 5-1020](#).

Return to the [Summary Table](#).

**Table 5-1020. PADAX\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.25 PADAY\_cfg\_reg Register (Offset = 60h) [Reset = 00000000h]

PADAY\_cfg\_reg is shown in [Table 5-1021](#).

Return to the [Summary Table](#).

**Table 5-1021. PADAY\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.26 PADAZ\_cfg\_reg Register (Offset = 64h) [Reset = 00000000h]

PADAZ\_cfg\_reg is shown in [Table 5-1022](#).

Return to the [Summary Table](#).

**Table 5-1022. PADAZ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.27 PADBA\_cfg\_reg Register (Offset = 68h) [Reset = 00000000h]

PADBA\_cfg\_reg is shown in [Table 5-1023](#).

Return to the [Summary Table](#).

**Table 5-1023. PADBA\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.28 PADBB\_cfg\_reg Register (Offset = 6Ch) [Reset = 00000000h]

PADBB\_cfg\_reg is shown in [Table 5-1024](#).

Return to the [Summary Table](#).

**Table 5-1024. PADBB\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.29 PADBC\_cfg\_reg Register (Offset = 70h) [Reset = 00000000h]

PADBC\_cfg\_reg is shown in [Table 5-1025](#).

Return to the [Summary Table](#).

**Table 5-1025. PADBC\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.30 PADBD\_cfg\_reg Register (Offset = 74h) [Reset = 00000000h]

PADBD\_cfg\_reg is shown in [Table 5-1026](#).

Return to the [Summary Table](#).

**Table 5-1026. PADBD\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.31 PADBE\_cfg\_reg Register (Offset = 78h) [Reset = 00000000h]

PADBE\_cfg\_reg is shown in [Table 5-1027](#).

Return to the [Summary Table](#).

**Table 5-1027. PADBE\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.32 PADBF\_cfg\_reg Register (Offset = 7Ch) [Reset = 00000000h]

PADBF\_cfg\_reg is shown in [Table 5-1028](#).

Return to the [Summary Table](#).

**Table 5-1028. PADBF\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.33 PADBG\_cfg\_reg Register (Offset = 80h) [Reset = 00000000h]

PADBG\_cfg\_reg is shown in [Table 5-1029](#).

Return to the [Summary Table](#).

**Table 5-1029. PADBG\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.34 PADBH\_cfg\_reg Register (Offset = 84h) [Reset = 00000000h]

PADBH\_cfg\_reg is shown in [Table 5-1030](#).

Return to the [Summary Table](#).

**Table 5-1030. PADBH\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.35 PADBI\_cfg\_reg Register (Offset = 88h) [Reset = 00000000h]

PADBI\_cfg\_reg is shown in [Table 5-1031](#).

Return to the [Summary Table](#).

**Table 5-1031. PADBI\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.36 PADBJ\_cfg\_reg Register (Offset = 8Ch) [Reset = 00000000h]

PADBJ\_cfg\_reg is shown in [Table 5-1032](#).

Return to the [Summary Table](#).

**Table 5-1032. PADBJ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.37 PADBK\_cfg\_reg Register (Offset = 90h) [Reset = 00000000h]

PADBK\_cfg\_reg is shown in [Table 5-1033](#).

Return to the [Summary Table](#).

**Table 5-1033. PADBK\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.38 PADBL\_cfg\_reg Register (Offset = 94h) [Reset = 00000000h]

PADBL\_cfg\_reg is shown in [Table 5-1034](#).

Return to the [Summary Table](#).

**Table 5-1034. PADBL\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.39 PADBM\_cfg\_reg Register (Offset = 98h) [Reset = 00000000h]

PADBM\_cfg\_reg is shown in [Table 5-1035](#).

Return to the [Summary Table](#).

**Table 5-1035. PADBM\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

#### 5.2.8.40 PADBN\_cfg\_reg Register (Offset = 9Ch) [Reset = 00000000h]

PADBN\_cfg\_reg is shown in [Table 5-1036](#).

Return to the [Summary Table](#).

**Table 5-1036. PADBN\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

#### 5.2.8.41 PADBO\_cfg\_reg Register (Offset = A0h) [Reset = 00000000h]

PADBO\_cfg\_reg is shown in [Table 5-1037](#).

Return to the [Summary Table](#).

**Table 5-1037. PADBO\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

#### 5.2.8.42 PADBP\_cfg\_reg Register (Offset = A4h) [Reset = 00000000h]

PADBP\_cfg\_reg is shown in [Table 5-1038](#).

Return to the [Summary Table](#).

**Table 5-1038. PADBP\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.43 PADBQ\_cfg\_reg Register (Offset = A8h) [Reset = 00000000h]

PADBQ\_cfg\_reg is shown in [Table 5-1039](#).

Return to the [Summary Table](#).

**Table 5-1039. PADBQ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



#### 5.2.8.44 PADBR\_cfg\_reg Register (Offset = ACh) [Reset = 00000000h]

PADBR\_cfg\_reg is shown in [Table 5-1040](#).

Return to the [Summary Table](#).

**Table 5-1040. PADBR\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.45 PADBS\_cfg\_reg Register (Offset = B0h) [Reset = 00000000h]

PADBS\_cfg\_reg is shown in [Table 5-1041](#).

Return to the [Summary Table](#).

**Table 5-1041. PADBS\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

#### 5.2.8.46 PADBT\_cfg\_reg Register (Offset = B4h) [Reset = 00000000h]

PADBT\_cfg\_reg is shown in [Table 5-1042](#).

Return to the [Summary Table](#).

**Table 5-1042. PADBT\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.47 PADBU\_cfg\_reg Register (Offset = B8h) [Reset = 00000000h]

PADBU\_cfg\_reg is shown in [Table 5-1043](#).

Return to the [Summary Table](#).

**Table 5-1043. PADBU\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.48 PADBV\_cfg\_reg Register (Offset = BCh) [Reset = 00000000h]

PADBV\_cfg\_reg is shown in [Table 5-1044](#).

Return to the [Summary Table](#).

**Table 5-1044. PADBV\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.49 PADBW\_cfg\_reg Register (Offset = C0h) [Reset = 00000000h]

PADBW\_cfg\_reg is shown in [Table 5-1045](#).

Return to the [Summary Table](#).

**Table 5-1045. PADBW\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.50 PADBX\_cfg\_reg Register (Offset = C4h) [Reset = 00000000h]

PADBX\_cfg\_reg is shown in [Table 5-1046](#).

Return to the [Summary Table](#).

**Table 5-1046. PADBX\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.51 PADBY\_cfg\_reg Register (Offset = C8h) [Reset = 00000000h]

PADBY\_cfg\_reg is shown in [Table 5-1047](#).

Return to the [Summary Table](#).

**Table 5-1047. PADBY\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.52 PADBZ\_cfg\_reg Register (Offset = CCh) [Reset = 00000000h]

PADBZ\_cfg\_reg is shown in [Table 5-1048](#).

Return to the [Summary Table](#).

**Table 5-1048. PADBZ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.53 PADCA\_cfg\_reg Register (Offset = D0h) [Reset = 00000000h]

PADCA\_cfg\_reg is shown in [Table 5-1049](#).

Return to the [Summary Table](#).

**Table 5-1049. PADCA\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.54 PADCB\_cfg\_reg Register (Offset = D4h) [Reset = 00000000h]

PADCB\_cfg\_reg is shown in [Table 5-1050](#).

Return to the [Summary Table](#).

**Table 5-1050. PADCB\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.55 PADCC\_cfg\_reg Register (Offset = D8h) [Reset = 00000000h]

PADCC\_cfg\_reg is shown in [Table 5-1051](#).

Return to the [Summary Table](#).

**Table 5-1051. PADCC\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.56 PADCD\_cfg\_reg Register (Offset = DCh) [Reset = 00000000h]

PADCD\_cfg\_reg is shown in [Table 5-1052](#).

Return to the [Summary Table](#).

**Table 5-1052. PADCD\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.57 PADCE\_cfg\_reg Register (Offset = E0h) [Reset = 00000000h]

PADCE\_cfg\_reg is shown in [Table 5-1053](#).

Return to the [Summary Table](#).

**Table 5-1053. PADCE\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.58 PADCF\_cfg\_reg Register (Offset = E4h) [Reset = 00000000h]

PADCF\_cfg\_reg is shown in [Table 5-1054](#).

Return to the [Summary Table](#).

**Table 5-1054. PADCF\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.59 PADCG\_cfg\_reg Register (Offset = E8h) [Reset = 00000000h]

PADCG\_cfg\_reg is shown in [Table 5-1055](#).

Return to the [Summary Table](#).

**Table 5-1055. PADCG\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.60 PADCH\_cfg\_reg Register (Offset = ECh) [Reset = 00000000h]

PADCH\_cfg\_reg is shown in [Table 5-1056](#).

Return to the [Summary Table](#).

**Table 5-1056. PADCH\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.61 PADCI\_cfg\_reg Register (Offset = F0h) [Reset = 00000000h]

PADCI\_cfg\_reg is shown in [Table 5-1057](#).

Return to the [Summary Table](#).

**Table 5-1057. PADCI\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.62 PADCJ\_cfg\_reg Register (Offset = F4h) [Reset = 00000000h]

PADCJ\_cfg\_reg is shown in [Table 5-1058](#).

Return to the [Summary Table](#).

**Table 5-1058. PADCJ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.63 PADCK\_cfg\_reg Register (Offset = F8h) [Reset = 00000000h]

PADCK\_cfg\_reg is shown in [Table 5-1059](#).

Return to the [Summary Table](#).

**Table 5-1059. PADCK\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.64 PADCL\_cfg\_reg Register (Offset = FCh) [Reset = 00000000h]

PADCL\_cfg\_reg is shown in [Table 5-1060](#).

Return to the [Summary Table](#).

**Table 5-1060. PADCL\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.65 PADCM\_cfg\_reg Register (Offset = 100h) [Reset = 00000000h]

PADCM\_cfg\_reg is shown in [Table 5-1061](#).

Return to the [Summary Table](#).

**Table 5-1061. PADCM\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.66 PADCN\_cfg\_reg Register (Offset = 104h) [Reset = 00000000h]

PADCN\_cfg\_reg is shown in [Table 5-1062](#).

Return to the [Summary Table](#).

**Table 5-1062. PADCN\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.67 PADCO\_cfg\_reg Register (Offset = 108h) [Reset = 00000000h]

PADCO\_cfg\_reg is shown in [Table 5-1063](#).

Return to the [Summary Table](#).

**Table 5-1063. PADCO\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.68 PADCP\_cfg\_reg Register (Offset = 10Ch) [Reset = 00000000h]

PADCP\_cfg\_reg is shown in [Table 5-1064](#).

Return to the [Summary Table](#).

**Table 5-1064. PADCP\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.69 PADQC\_cfg\_reg Register (Offset = 110h) [Reset = 00000000h]

PADCQ\_cfg\_reg is shown in [Table 5-1065](#).

Return to the [Summary Table](#).

**Table 5-1065. PADQC\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.70 PADCR\_cfg\_reg Register (Offset = 114h) [Reset = 00000000h]

PADCR\_cfg\_reg is shown in [Table 5-1066](#).

Return to the [Summary Table](#).

**Table 5-1066. PADCR\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.71 PADCS\_cfg\_reg Register (Offset = 118h) [Reset = 00000000h]

PADCS\_cfg\_reg is shown in [Table 5-1067](#).

Return to the [Summary Table](#).

**Table 5-1067. PADCS\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.72 PADCT\_cfg\_reg Register (Offset = 11Ch) [Reset = 00000000h]

PADCT\_cfg\_reg is shown in [Table 5-1068](#).

Return to the [Summary Table](#).

**Table 5-1068. PADCT\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.73 PADCU\_cfg\_reg Register (Offset = 120h) [Reset = 00000000h]

PADCU\_cfg\_reg is shown in [Table 5-1069](#).

Return to the [Summary Table](#).

**Table 5-1069. PADCU\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

#### 5.2.8.74 PADCV\_cfg\_reg Register (Offset = 124h) [Reset = 00000000h]

PADCV\_cfg\_reg is shown in [Table 5-1070](#).

Return to the [Summary Table](#).

**Table 5-1070. PADCV\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.75 PADCW\_cfg\_reg Register (Offset = 128h) [Reset = 00000000h]

PADCW\_cfg\_reg is shown in [Table 5-1071](#).

Return to the [Summary Table](#).

**Table 5-1071. PADCW\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.76 PADCX\_cfg\_reg Register (Offset = 12Ch) [Reset = 00000000h]

PADCX\_cfg\_reg is shown in [Table 5-1072](#).

Return to the [Summary Table](#).

**Table 5-1072. PADCX\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.77 PADCY\_cfg\_reg Register (Offset = 130h) [Reset = 00000000h]

PADCY\_cfg\_reg is shown in [Table 5-1073](#).

Return to the [Summary Table](#).

**Table 5-1073. PADCY\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.78 PADCZ\_cfg\_reg Register (Offset = 134h) [Reset = 00000000h]

PADCZ\_cfg\_reg is shown in [Table 5-1074](#).

Return to the [Summary Table](#).

**Table 5-1074. PADCZ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.79 PADDA\_cfg\_reg Register (Offset = 138h) [Reset = 00000000h]

PADDA\_cfg\_reg is shown in [Table 5-1075](#).

Return to the [Summary Table](#).

**Table 5-1075. PADDA\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.80 PADDB\_cfg\_reg Register (Offset = 13Ch) [Reset = 00000000h]

PADDB\_cfg\_reg is shown in [Table 5-1076](#).

Return to the [Summary Table](#).

**Table 5-1076. PADDB\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.81 PADDC\_cfg\_reg Register (Offset = 140h) [Reset = 00000000h]

PADDC\_cfg\_reg is shown in [Table 5-1077](#).

Return to the [Summary Table](#).

**Table 5-1077. PADDC\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.82 PADDD\_cfg\_reg Register (Offset = 144h) [Reset = 00000000h]

PADDD\_cfg\_reg is shown in [Table 5-1078](#).

Return to the [Summary Table](#).

**Table 5-1078. PADDD\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.83 PADDE\_cfg\_reg Register (Offset = 148h) [Reset = 00000000h]

PADDE\_cfg\_reg is shown in [Table 5-1079](#).

Return to the [Summary Table](#).

**Table 5-1079. PADDE\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.84 PADDF\_cfg\_reg Register (Offset = 14Ch) [Reset = 00000000h]

PADDF\_cfg\_reg is shown in [Table 5-1080](#).

Return to the [Summary Table](#).

**Table 5-1080. PADDF\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.85 PADDG\_cfg\_reg Register (Offset = 150h) [Reset = 00000000h]

PADDG\_cfg\_reg is shown in [Table 5-1081](#).

Return to the [Summary Table](#).

**Table 5-1081. PADDG\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.86 PADDH\_cfg\_reg Register (Offset = 154h) [Reset = 00000000h]

PADDH\_cfg\_reg is shown in [Table 5-1082](#).

Return to the [Summary Table](#).

**Table 5-1082. PADDH\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.87 PADDI\_cfg\_reg Register (Offset = 158h) [Reset = 00000000h]

PADDI\_cfg\_reg is shown in [Table 5-1083](#).

Return to the [Summary Table](#).

**Table 5-1083. PADDI\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.88 PADDJ\_cfg\_reg Register (Offset = 15Ch) [Reset = 00000000h]

PADDJ\_cfg\_reg is shown in [Table 5-1084](#).

Return to the [Summary Table](#).

**Table 5-1084. PADDJ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.89 PADDK\_cfg\_reg Register (Offset = 160h) [Reset = 00000000h]

PADDK\_cfg\_reg is shown in [Table 5-1085](#).

Return to the [Summary Table](#).

**Table 5-1085. PADDK\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.90 PADDL\_cfg\_reg Register (Offset = 164h) [Reset = 00000000h]

PADDL\_cfg\_reg is shown in [Table 5-1086](#).

Return to the [Summary Table](#).

**Table 5-1086. PADDL\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.91 PADDM\_cfg\_reg Register (Offset = 168h) [Reset = 00000000h]

PADDM\_cfg\_reg is shown in [Table 5-1087](#).

Return to the [Summary Table](#).

**Table 5-1087. PADDM\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.92 PADDN\_cfg\_reg Register (Offset = 16Ch) [Reset = 00000000h]

PADDN\_cfg\_reg is shown in [Table 5-1088](#).

Return to the [Summary Table](#).

**Table 5-1088. PADDN\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.93 PADD0\_cfg\_reg Register (Offset = 170h) [Reset = 00000000h]

PADD0\_cfg\_reg is shown in [Table 5-1089](#).

Return to the [Summary Table](#).

**Table 5-1089. PADD0\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.94 PADDP\_cfg\_reg Register (Offset = 174h) [Reset = 00000000h]

PADDP\_cfg\_reg is shown in [Table 5-1090](#).

Return to the [Summary Table](#).

**Table 5-1090. PADDP\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.95 PADDQ\_cfg\_reg Register (Offset = 178h) [Reset = 00000000h]

PADDQ\_cfg\_reg is shown in [Table 5-1091](#).

Return to the [Summary Table](#).

**Table 5-1091. PADDQ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.96 PADDR\_cfg\_reg Register (Offset = 17Ch) [Reset = 00000000h]

PADDR\_cfg\_reg is shown in [Table 5-1092](#).

Return to the [Summary Table](#).

**Table 5-1092. PADDR\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.97 PADDs\_cfg\_reg Register (Offset = 180h) [Reset = 00000000h]

PADDs\_cfg\_reg is shown in [Table 5-1093](#).

Return to the [Summary Table](#).

**Table 5-1093. PADDs\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.98 PADDT\_cfg\_reg Register (Offset = 184h) [Reset = 00000000h]

PADDT\_cfg\_reg is shown in [Table 5-1094](#).

Return to the [Summary Table](#).

**Table 5-1094. PADDT\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.99 PADDU\_cfg\_reg Register (Offset = 188h) [Reset = 00000000h]

PADDU\_cfg\_reg is shown in [Table 5-1095](#).

Return to the [Summary Table](#).

**Table 5-1095. PADDU\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.100 PADDV\_cfg\_reg Register (Offset = 18Ch) [Reset = 00000000h]

PADDV\_cfg\_reg is shown in [Table 5-1096](#).

Return to the [Summary Table](#).

**Table 5-1096. PADDV\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.101 PADDW\_cfg\_reg Register (Offset = 190h) [Reset = 00000000h]

PADDW\_cfg\_reg is shown in [Table 5-1097](#).

Return to the [Summary Table](#).

**Table 5-1097. PADDW\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.102 PADDX\_cfg\_reg Register (Offset = 194h) [Reset = 00000000h]

PADDX\_cfg\_reg is shown in [Table 5-1098](#).

Return to the [Summary Table](#).

**Table 5-1098. PADDX\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.103 PADDY\_cfg\_reg Register (Offset = 198h) [Reset = 00000000h]

PADDY\_cfg\_reg is shown in [Table 5-1099](#).

Return to the [Summary Table](#).

**Table 5-1099. PADDY\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.104 PADDZ\_cfg\_reg Register (Offset = 19Ch) [Reset = 00000000h]

PADDZ\_cfg\_reg is shown in [Table 5-1100](#).

Return to the [Summary Table](#).

**Table 5-1100. PADDZ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.105 PADEA\_cfg\_reg Register (Offset = 1A0h) [Reset = 00000000h]

PADEA\_cfg\_reg is shown in [Table 5-1101](#).

Return to the [Summary Table](#).

**Table 5-1101. PADEA\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.106 PADEB\_cfg\_reg Register (Offset = 1A4h) [Reset = 00000000h]

PADEB\_cfg\_reg is shown in [Table 5-1102](#).

Return to the [Summary Table](#).

**Table 5-1102. PADEB\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.107 PADEC\_cfg\_reg Register (Offset = 1A8h) [Reset = 00000000h]

PADEC\_cfg\_reg is shown in [Table 5-1103](#).

Return to the [Summary Table](#).

**Table 5-1103. PADEC\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.108 PADED\_cfg\_reg Register (Offset = 1ACh) [Reset = 00000000h]

PADED\_cfg\_reg is shown in [Table 5-1104](#).

Return to the [Summary Table](#).

**Table 5-1104. PADED\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.109 PADEE\_cfg\_reg Register (Offset = 1B0h) [Reset = 00000000h]

PADEE\_cfg\_reg is shown in [Table 5-1105](#).

Return to the [Summary Table](#).

**Table 5-1105. PADEE\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.110 PADEF\_cfg\_reg Register (Offset = 1B4h) [Reset = 00000000h]

PADEF\_cfg\_reg is shown in [Table 5-1106](#).

Return to the [Summary Table](#).

**Table 5-1106. PADEF\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.111 PADEG\_cfg\_reg Register (Offset = 1B8h) [Reset = 00000000h]

PADEG\_cfg\_reg is shown in [Table 5-1107](#).

Return to the [Summary Table](#).

**Table 5-1107. PADEG\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.112 PADEH\_cfg\_reg Register (Offset = 1BCh) [Reset = 00000000h]

PADEH\_cfg\_reg is shown in [Table 5-1108](#).

Return to the [Summary Table](#).

**Table 5-1108. PADEH\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.113 PADEI\_cfg\_reg Register (Offset = 1C0h) [Reset = 00000000h]

PADEI\_cfg\_reg is shown in [Table 5-1109](#).

Return to the [Summary Table](#).

**Table 5-1109. PADEI\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.114 PADEJ\_cfg\_reg Register (Offset = 1C4h) [Reset = 00000000h]

PADEJ\_cfg\_reg is shown in [Table 5-1110](#).

Return to the [Summary Table](#).

**Table 5-1110. PADEJ\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.115 PADEK\_cfg\_reg Register (Offset = 1C8h) [Reset = 00000000h]

PADEK\_cfg\_reg is shown in [Table 5-1111](#).

Return to the [Summary Table](#).

**Table 5-1111. PADEK\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select



### 5.2.8.116 PADEL\_cfg\_reg Register (Offset = 1CCh) [Reset = 00000000h]

PADEL\_cfg\_reg is shown in [Table 5-1112](#).

Return to the [Summary Table](#).

**Table 5-1112. PADEL\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.117 PADEM\_cfg\_reg Register (Offset = 1D0h) [Reset = 00000000h]

PADEM\_cfg\_reg is shown in [Table 5-1113](#).

Return to the [Summary Table](#).

**Table 5-1113. PADEM\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.118 PADEN\_cfg\_reg Register (Offset = 1D4h) [Reset = 00000000h]

PADEN\_cfg\_reg is shown in [Table 5-1114](#).

Return to the [Summary Table](#).

**Table 5-1114. PADEN\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.119 PADEO\_cfg\_reg Register (Offset = 1D8h) [Reset = 00000000h]

PADEO\_cfg\_reg is shown in [Table 5-1115](#).

Return to the [Summary Table](#).

**Table 5-1115. PADEO\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.120 PADEP\_cfg\_reg Register (Offset = 1DCh) [Reset = 00000000h]

PADEP\_cfg\_reg is shown in [Table 5-1116](#).

Return to the [Summary Table](#).

**Table 5-1116. PADEP\_cfg\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	NU	R/W	0h	Reserved
10	sc1	R/W	0h	IO Slew rate control : 0 : higher slew rate. 1: Lower slew rate.
9	pupdsel	R/W	0h	Pullup/PullDown Selection 0 -- Pull Down
8	pi	R/W	0h	Pull Inhibit/Pull Disable 0 -- Enable
7	oe_override	R/W	0h	Active Low Output Override
6	oe_override_ctrl	R/W	0h	Active Low Output Override Control : Write 1 to select Active low Output Override value to control IOs OE_N/GZ instead of the control from hardware
5	ie_override	R/W	0h	Active Low Input Override
4	ie_override_ctrl	R/W	0h	Active Low Input Override Control : Write 1 to select Active low Input Override value to control IOs IE_N/RXACTIVE_N instead of the control from hardware
3-0	func_sel	R/W	0h	Function select

### 5.2.8.121 USERMODEEN Register (Offset = 1F0h) [Reset = 00000000h]

USERMODEEN is shown in [Table 5-1117](#).

Return to the [Summary Table](#).

**Table 5-1117. USERMODEEN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	USERMODEEN	R/W	0h	Write 0XADADADAD to enable user mode write access to IO CFG space

### 5.2.8.122 PADGLBLCFGREG Register (Offset = 1F4h) [Reset = 00000000h]

PADGLBLCFGREG is shown in [Table 5-1118](#).

Return to the [Summary Table](#).

**Table 5-1118. PADGLBLCFGREG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PADGLBLCFGREG	R/W	0h	2:0 : global_ie_n_ctl - Write 3'b111 to pass global_ie_n_val to IE_N/ RXACTIVE_N pin of all the IOs. 3 : global_ie_n_val - Active low 10:8 : global_oe_n_ctl - Write 3'b111 to pass global_oe_n_val to OE_N/GZ pin of all the IOs. 11 : global_oe_n_val - Active low 18:16 : global_pi_ctl - Write 3'b111 to pass global_pi_val and global_pu_val to all the IOs 19 : global_pi_val 20 : global_pu_val

### 5.2.8.123 IOCFGKICK0 Register (Offset = 1F8h) [Reset = 00000000h]

IOCFGKICK0 is shown in [Table 5-1119](#).

Return to the [Summary Table](#).

**Table 5-1119. IOCFGKICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IOCFGKICK0	R/W	0h	Kicker 0 Register. The value 83E7 0B13h must be written to KICK0 as part of the process to unlock the CPU write access to the above PIN MUX registers (including IOCFGKICK1)



### 5.2.8.124 IOCFGKICK1 Register (Offset = 1FCh) [Reset = 00000000h]

IOCFGKICK1 is shown in [Table 5-1120](#).

Return to the [Summary Table](#).

**Table 5-1120. IOCFGKICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IOCFGKICK1	R/W	0h	Kicker 1 Register. The value 95A4 F1E0h must be written to the KICK1 as part of the process to unlock the CPU write access to above PINMUX registers (excluding IOCFGKICK0). IOCFGKICK0 has to be written with 83E70B13h to enable access to IOCFGKICK1.

### 5.2.9 TOP\_CTRL Registers

Table 5-1121 lists the memory-mapped registers for the TOP\_CTRL registers. All register offset addresses not listed in Table 5-1121 should be considered as reserved locations and the register contents should not be modified.

**Table 5-1121. TOP\_CTRL Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.9.1</a>
4h	MDO_CTRL		<a href="#">Section 5.2.9.2</a>
8h	PROBE_BUS_SELO		<a href="#">Section 5.2.9.3</a>
Ch	PROBE_BUS_SEL1		<a href="#">Section 5.2.9.4</a>
10h	RS232_SLEEP_CLK_DIV		<a href="#">Section 5.2.9.5</a>
14h	RS232_SLEEP_CLK_DIV_by2		<a href="#">Section 5.2.9.6</a>
18h	RS232_STATUS		<a href="#">Section 5.2.9.7</a>
200h	EFUSE_DIEID0		<a href="#">Section 5.2.9.8</a>
204h	EFUSE_DIEID1		<a href="#">Section 5.2.9.9</a>
208h	EFUSE_DIEID2		<a href="#">Section 5.2.9.10</a>
20Ch	EFUSE_DIEID3		<a href="#">Section 5.2.9.11</a>
210h	EFUSE_UID0		<a href="#">Section 5.2.9.12</a>
214h	EFUSE_UID1		<a href="#">Section 5.2.9.13</a>
218h	EFUSE_UID2		<a href="#">Section 5.2.9.14</a>
21Ch	EFUSE_UID3		<a href="#">Section 5.2.9.15</a>
220h	PREVIOUS_NAME		<a href="#">Section 5.2.9.16</a>
224h	EFUSE_FROM0_CHECKSUM		<a href="#">Section 5.2.9.17</a>
228h	EFUSE_ROM_SEQ_UPDATE0		<a href="#">Section 5.2.9.18</a>
22Ch	EFUSE_ROM_SEQ_UPDATE1		<a href="#">Section 5.2.9.19</a>
230h	EFUSE_ROM_SEQ_UPDATE2		<a href="#">Section 5.2.9.20</a>
234h	EFUSE_ROM_SEQ_UPDATE3		<a href="#">Section 5.2.9.21</a>
238h	EFUSE_ROM_SEQ_UPDATE4		<a href="#">Section 5.2.9.22</a>
23Ch	EFUSE_ROM_SEQ_UPDATE5		<a href="#">Section 5.2.9.23</a>
240h	EFUSE_ROM_SEQ_UPDATE6		<a href="#">Section 5.2.9.24</a>
244h	EFUSE_ROM_SEQ_UPDATE7		<a href="#">Section 5.2.9.25</a>
248h	EFUSE_ROM_SEQ_UPDATE8		<a href="#">Section 5.2.9.26</a>
400h	EFUSE0_ROW_61		<a href="#">Section 5.2.9.27</a>
404h	EFUSE0_ROW_62		<a href="#">Section 5.2.9.28</a>
408h	EFUSE0_ROW_63		<a href="#">Section 5.2.9.29</a>
40Ch	EFUSE1_ROW_5		<a href="#">Section 5.2.9.30</a>
410h	EFUSE1_ROW_6		<a href="#">Section 5.2.9.31</a>
414h	EFUSE1_ROW_7		<a href="#">Section 5.2.9.32</a>
418h	EFUSE1_ROW_8		<a href="#">Section 5.2.9.33</a>
41Ch	EFUSE1_ROW_9		<a href="#">Section 5.2.9.34</a>
420h	EFUSE1_ROW_10		<a href="#">Section 5.2.9.35</a>
424h	EFUSE1_ROW_11		<a href="#">Section 5.2.9.36</a>
428h	EFUSE1_ROW_12		<a href="#">Section 5.2.9.37</a>
42Ch	EFUSE1_ROW_13		<a href="#">Section 5.2.9.38</a>
430h	EFUSE1_ROW_14		<a href="#">Section 5.2.9.39</a>
434h	EFUSE1_ROW_15		<a href="#">Section 5.2.9.40</a>

**Table 5-1121. TOP\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
438h	EFUSE1_ROW_16		<a href="#">Section 5.2.9.41</a>
43Ch	EFUSE1_ROW_17		<a href="#">Section 5.2.9.42</a>
440h	EFUSE1_ROW_18		<a href="#">Section 5.2.9.43</a>
444h	EFUSE1_ROW_19		<a href="#">Section 5.2.9.44</a>
448h	EFUSE1_ROW_20		<a href="#">Section 5.2.9.45</a>
44Ch	EFUSE1_ROW_21		<a href="#">Section 5.2.9.46</a>
450h	EFUSE1_ROW_22		<a href="#">Section 5.2.9.47</a>
454h	EFUSE1_ROW_23		<a href="#">Section 5.2.9.48</a>
458h	EFUSE1_ROW_24		<a href="#">Section 5.2.9.49</a>
45Ch	EFUSE1_ROW_25		<a href="#">Section 5.2.9.50</a>
460h	EFUSE1_ROW_26		<a href="#">Section 5.2.9.51</a>
464h	EFUSE1_ROW_27		<a href="#">Section 5.2.9.52</a>
468h	EFUSE1_ROW_28		<a href="#">Section 5.2.9.53</a>
46Ch	EFUSE1_ROW_29		<a href="#">Section 5.2.9.54</a>
470h	EFUSE1_ROW_30		<a href="#">Section 5.2.9.55</a>
474h	EFUSE1_ROW_31		<a href="#">Section 5.2.9.56</a>
478h	EFUSE1_ROW_32		<a href="#">Section 5.2.9.57</a>
47Ch	EFUSE1_ROW_33		<a href="#">Section 5.2.9.58</a>
480h	EFUSE1_ROW_34		<a href="#">Section 5.2.9.59</a>
484h	EFUSE1_ROW_35		<a href="#">Section 5.2.9.60</a>
488h	EFUSE1_ROW_36		<a href="#">Section 5.2.9.61</a>
48Ch	EFUSE1_ROW_37		<a href="#">Section 5.2.9.62</a>
490h	EFUSE1_ROW_38		<a href="#">Section 5.2.9.63</a>
494h	EFUSE1_ROW_39		<a href="#">Section 5.2.9.64</a>
498h	EFUSE1_ROW_40		<a href="#">Section 5.2.9.65</a>
49Ch	EFUSE1_ROW_41		<a href="#">Section 5.2.9.66</a>
4A0h	EFUSE1_ROW_42		<a href="#">Section 5.2.9.67</a>
4A4h	EFUSE1_ROW_43		<a href="#">Section 5.2.9.68</a>
800h	EFUSE_OVERRIDE_HSM_HALT_ON_ROM_ECC_ERR_EN		<a href="#">Section 5.2.9.69</a>
804h	EFUSE_OVERRIDE_MEM_MARGINCTRL		<a href="#">Section 5.2.9.70</a>
808h	EFUSE_OVERRIDE_LVDS_BGAP_TRIM		<a href="#">Section 5.2.9.71</a>
80Ch	EFUSE_OVERRIDE_XTAL_STABLIZATION_WAIT		<a href="#">Section 5.2.9.72</a>
810h	EFUSE_OVERRIDE_SLICER_BIAS_RTRIM		<a href="#">Section 5.2.9.73</a>
814h	EFUSE_OVERRIDE_XO_OUTPUT_DRIVE		<a href="#">Section 5.2.9.74</a>
818h	EFUSE_OVERRIDE_RCOSC_TRIM_CODE		<a href="#">Section 5.2.9.75</a>
81Ch	EFUSE_OVERRIDE_IP1_BG1_RTRIM		<a href="#">Section 5.2.9.76</a>
820h	EFUSE_OVERRIDE_IP1_BG1_SLOPE		<a href="#">Section 5.2.9.77</a>
824h	EFUSE_OVERRIDE_IP1_BG1_MAG		<a href="#">Section 5.2.9.78</a>
828h	EFUSE_OVERRIDE_RS232_CLKMODE		<a href="#">Section 5.2.9.79</a>
82Ch	EFUSE_OVERRIDE_VMON_VDD_OV_UV_TRIM		<a href="#">Section 5.2.9.80</a>

**Table 5-1121. TOP\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
830h	EFUSE_OVERRIDE_VMON_VDDS_3P3_UV_TRIM		<a href="#">Section 5.2.9.81</a>
834h	EFUSE_OVERRIDE_VMON_VDDA_OSC_TRIM		<a href="#">Section 5.2.9.82</a>
838h	EFUSE_OVERRIDE_VDD_VT_DET		<a href="#">Section 5.2.9.83</a>
83Ch	EFUSE_OVERRIDE_MASK_CPU_CLK_O UT_CTRL_LOWV_VAL		<a href="#">Section 5.2.9.84</a>
840h	EFUSE_OVERRIDE_MASK_CPU_CLK_O UT_CTRL_LOWV_SEL		<a href="#">Section 5.2.9.85</a>
844h	EFUSE_OVERRIDE_EN_VOL_MON_FUNC		<a href="#">Section 5.2.9.86</a>
848h	EFUSE_OVERRIDE_EN_VOL_MON_FUNC		<a href="#">Section 5.2.9.86</a>
84Ch	EFUSE_OVERRIDE_SPARE_ANA		<a href="#">Section 5.2.9.87</a>
850h	EFUSE_OVERRIDE_SLICER_DLY_DISABLE		<a href="#">Section 5.2.9.88</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.9.89</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.9.90</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.9.91</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.9.92</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.9.93</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.9.94</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.9.95</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.9.96</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.9.97</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.9.98</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.9.99</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.9.100</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.9.101</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.9.102</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.9.103</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.9.104</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.9.105</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.9.106</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.9.107</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.9.108</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.9.109</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-1122](#) shows the codes that are used for access types in this section.

**Table 5-1122. TOP\_CTRL Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write

**Table 5-1122. TOP\_CTRL Access Type Codes  
(continued)**

Access Type	Code	Description
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.9.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-1123](#).

Return to the [Summary Table](#).

PID register

**Table 5-1123. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.9.2 MDO\_CTRL Register (Offset = 4h) [Reset = 0000000Xh]

MDO\_CTRL is shown in [Table 5-1124](#).

Return to the [Summary Table](#).

**Table 5-1124. MDO\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	
5-4	SRC_SELECT	R/W	0h	Select the source IP of LVDS Data 0: Aurora full data on LVDS 1: CBUFF on LVDS 2: Aurora bit-clk ,frame clk and 2 data support. 3: Aurora 1bit clk and 2 data support for LOP package
3-1	RESERVED	R/W	0h	
0	AURORATX_SRC_SELECT	R/W	0h	Select the TPIU source to TOP_AURORATX IP 0:Measurement Data 1: Trace Data

### 5.2.9.3 PROBE\_BUS\_SEL0 Register (Offset = 8h) [Reset = 00000000h]

PROBE\_BUS\_SEL0 is shown in [Table 5-1125](#).

Return to the [Summary Table](#).

**Table 5-1125. PROBE\_BUS\_SEL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	sel	R/W	0h	Probe Bus 0 Mux Select



#### 5.2.9.4 PROBE\_BUS\_SEL1 Register (Offset = Ch) [Reset = 00000000h]

PROBE\_BUS\_SEL1 is shown in [Table 5-1126](#).

Return to the [Summary Table](#).

**Table 5-1126. PROBE\_BUS\_SEL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	sel	R/W	0h	Probe Bus 1 Mux Select

### 5.2.9.5 RS232\_SLEEP\_CLK\_DIV Register (Offset = 10h) [Reset = 00000131h]

RS232\_SLEEP\_CLK\_DIV is shown in [Table 5-1127](#).

Return to the [Summary Table](#).

**Table 5-1127. RS232\_SLEEP\_CLK\_DIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	div_val	R/W	131h	The Divider value for RS232 sleep clock generation from RCclk

### 5.2.9.6 RS232\_SLEEP\_CLK\_DIV\_by2 Register (Offset = 14h) [Reset = 00000098h]

RS232\_SLEEP\_CLK\_DIV\_by2 is shown in [Table 5-1128](#).

Return to the [Summary Table](#).

**Table 5-1128. RS232\_SLEEP\_CLK\_DIV\_by2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	div_val	R/W	98h	The Dividerby2 value for RS232 sleep clock generation from RCclk. This decides the duty cycle of sleep clock.

### 5.2.9.7 RS232\_STATUS Register (Offset = 18h) [Reset = 02B6X000h]

RS232\_STATUS is shown in [Table 5-1129](#).

Return to the [Summary Table](#).

**Table 5-1129. RS232\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-16	curr_bit_interval	R	2B6h	Current RS232 bit interval being used
15-12	RESERVED	R	0h	
11-0	clk_duration	R	0h	Number of vbusp clocks measured in one sleep clock frequency

### 5.2.9.8 EFUSE\_DIEID0 Register (Offset = 200h) [Reset = 00000000h]

EFUSE\_DIEID0 is shown in [Table 5-1130](#).

Return to the [Summary Table](#).

**Table 5-1130. EFUSE\_DIEID0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE DieID[31:0]

### 5.2.9.9 EFUSE\_DIEID1 Register (Offset = 204h) [Reset = 00000000h]

EFUSE\_DIEID1 is shown in [Table 5-1131](#).

Return to the [Summary Table](#).

**Table 5-1131. EFUSE\_DIEID1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE DieID[63:32]

### 5.2.9.10 EFUSE\_DIEID2 Register (Offset = 208h) [Reset = 00000000h]

EFUSE\_DIEID2 is shown in [Table 5-1132](#).

Return to the [Summary Table](#).

**Table 5-1132. EFUSE\_DIEID2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE DieID[95:64]

### 5.2.9.11 EFUSE\_DIEID3 Register (Offset = 20Ch) [Reset = 00000000h]

EFUSE\_DIEID3 is shown in [Table 5-1133](#).

Return to the [Summary Table](#).

**Table 5-1133. EFUSE\_DIEID3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE DieID[127:96]



### 5.2.9.12 EFUSE\_UID0 Register (Offset = 210h) [Reset = 00000000h]

EFUSE\_UID0 is shown in [Table 5-1134](#).

Return to the [Summary Table](#).

**Table 5-1134. EFUSE\_UID0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE UID[31:0]

### 5.2.9.13 EFUSE\_UID1 Register (Offset = 214h) [Reset = 00000000h]

EFUSE\_UID1 is shown in [Table 5-1135](#).

Return to the [Summary Table](#).

**Table 5-1135. EFUSE\_UID1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE UID[63:32]

#### 5.2.9.14 EFUSE\_UID2 Register (Offset = 218h) [Reset = 00000000h]

EFUSE\_UID2 is shown in [Table 5-1136](#).

Return to the [Summary Table](#).

**Table 5-1136. EFUSE\_UID2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE UID[95:64]

### 5.2.9.15 EFUSE\_UID3 Register (Offset = 21Ch) [Reset = 00000000h]

EFUSE\_UID3 is shown in [Table 5-1137](#).

Return to the [Summary Table](#).

**Table 5-1137. EFUSE\_UID3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	val	R	0h	EFUSE UID[120:96]

### 5.2.9.16 PREVIOUS\_NAME Register (Offset = 220h) [Reset = 00000000h]

PREVIOUS\_NAME is shown in [Table 5-1138](#).

Return to the [Summary Table](#).

**Table 5-1138. PREVIOUS\_NAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	val	R	0h	EFUSE Device Type

### 5.2.9.17 EFUSE\_FROM0\_CHECKSUM Register (Offset = 224h) [Reset = 00000000h]

EFUSE\_FROM0\_CHECKSUM is shown in [Table 5-1139](#).

Return to the [Summary Table](#).

**Table 5-1139. EFUSE\_FROM0\_CHECKSUM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	32 bit FROM0 Checksum

### 5.2.9.18 EFUSE\_ROM\_SEQ\_UPDATE0 Register (Offset = 228h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE0 is shown in [Table 5-1140](#).

Return to the [Summary Table](#).

**Table 5-1140. EFUSE\_ROM\_SEQ\_UPDATE0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update [31:0]

### 5.2.9.19 EFUSE\_ROM\_SEQ\_UPDATE1 Register (Offset = 22Ch) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE1 is shown in [Table 5-1141](#).

Return to the [Summary Table](#).

**Table 5-1141. EFUSE\_ROM\_SEQ\_UPDATE1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[63:32]



### 5.2.9.20 EFUSE\_ROM\_SEQ\_UPDATE2 Register (Offset = 230h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE2 is shown in [Table 5-1142](#).

Return to the [Summary Table](#).

**Table 5-1142. EFUSE\_ROM\_SEQ\_UPDATE2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[95:64]

### 5.2.9.21 EFUSE\_ROM\_SEQ\_UPDATE3 Register (Offset = 234h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE3 is shown in [Table 5-1143](#).

Return to the [Summary Table](#).

**Table 5-1143. EFUSE\_ROM\_SEQ\_UPDATE3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[127:96]

### 5.2.9.22 EFUSE\_ROM\_SEQ\_UPDATE4 Register (Offset = 238h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE4 is shown in [Table 5-1144](#).

Return to the [Summary Table](#).

**Table 5-1144. EFUSE\_ROM\_SEQ\_UPDATE4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[159:128]

### 5.2.9.23 EFUSE\_ROM\_SEQ\_UPDATE5 Register (Offset = 23Ch) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE5 is shown in [Table 5-1145](#).

Return to the [Summary Table](#).

**Table 5-1145. EFUSE\_ROM\_SEQ\_UPDATE5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[191:160]

#### 5.2.9.24 EFUSE\_ROM\_SEQ\_UPDATE6 Register (Offset = 240h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE6 is shown in [Table 5-1146](#).

Return to the [Summary Table](#).

**Table 5-1146. EFUSE\_ROM\_SEQ\_UPDATE6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[223:192]

### 5.2.9.25 EFUSE\_ROM\_SEQ\_UPDATE7 Register (Offset = 244h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE7 is shown in [Table 5-1147](#).

Return to the [Summary Table](#).

**Table 5-1147. EFUSE\_ROM\_SEQ\_UPDATE7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[255:224]

### 5.2.9.26 EFUSE\_ROM\_SEQ\_UPDATE8 Register (Offset = 248h) [Reset = 00000000h]

EFUSE\_ROM\_SEQ\_UPDATE8 is shown in [Table 5-1148](#).

Return to the [Summary Table](#).

**Table 5-1148. EFUSE\_ROM\_SEQ\_UPDATE8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	val	R	0h	EFUSE ROM Seq Update[287:256]

### 5.2.9.27 EFUSE0\_ROW\_61 Register (Offset = 400h) [Reset = 00000000h]

EFUSE0\_ROW\_61 is shown in [Table 5-1149](#).

Return to the [Summary Table](#).

**Table 5-1149. EFUSE0\_ROW\_61 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE0_ROW_61	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details



### 5.2.9.28 EFUSE0\_ROW\_62 Register (Offset = 404h) [Reset = 00000000h]

EFUSE0\_ROW\_62 is shown in [Table 5-1150](#).

Return to the [Summary Table](#).

**Table 5-1150. EFUSE0\_ROW\_62 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE0_ROW_62	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.29 EFUSE0\_ROW\_63 Register (Offset = 408h) [Reset = 00000000h]

EFUSE0\_ROW\_63 is shown in [Table 5-1151](#).

Return to the [Summary Table](#).

**Table 5-1151. EFUSE0\_ROW\_63 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE0_ROW_63	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.30 EFUSE1\_ROW\_5 Register (Offset = 40Ch) [Reset = 00000000h]

EFUSE1\_ROW\_5 is shown in [Table 5-1152](#).

Return to the [Summary Table](#).

**Table 5-1152. EFUSE1\_ROW\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_5	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.31 EFUSE1\_ROW\_6 Register (Offset = 410h) [Reset = 00000000h]

EFUSE1\_ROW\_6 is shown in [Table 5-1153](#).

Return to the [Summary Table](#).

**Table 5-1153. EFUSE1\_ROW\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_6	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.32 EFUSE1\_ROW\_7 Register (Offset = 414h) [Reset = 00000000h]

EFUSE1\_ROW\_7 is shown in [Table 5-1154](#).

Return to the [Summary Table](#).

**Table 5-1154. EFUSE1\_ROW\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_7	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.33 EFUSE1\_ROW\_8 Register (Offset = 418h) [Reset = 00000000h]

EFUSE1\_ROW\_8 is shown in [Table 5-1155](#).

Return to the [Summary Table](#).

**Table 5-1155. EFUSE1\_ROW\_8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_8	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.34 EFUSE1\_ROW\_9 Register (Offset = 41Ch) [Reset = 00000000h]

EFUSE1\_ROW\_9 is shown in [Table 5-1156](#).

Return to the [Summary Table](#).

**Table 5-1156. EFUSE1\_ROW\_9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_9	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.35 EFUSE1\_ROW\_10 Register (Offset = 420h) [Reset = 00000000h]

EFUSE1\_ROW\_10 is shown in [Table 5-1157](#).

Return to the [Summary Table](#).

**Table 5-1157. EFUSE1\_ROW\_10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_10	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details



### 5.2.9.36 EFUSE1\_ROW\_11 Register (Offset = 424h) [Reset = 00000000h]

EFUSE1\_ROW\_11 is shown in [Table 5-1158](#).

Return to the [Summary Table](#).

**Table 5-1158. EFUSE1\_ROW\_11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_11	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.37 EFUSE1\_ROW\_12 Register (Offset = 428h) [Reset = 00000000h]

EFUSE1\_ROW\_12 is shown in [Table 5-1159](#).

Return to the [Summary Table](#).

**Table 5-1159. EFUSE1\_ROW\_12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_12	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.38 EFUSE1\_ROW\_13 Register (Offset = 42Ch) [Reset = 00000000h]

EFUSE1\_ROW\_13 is shown in [Table 5-1160](#).

Return to the [Summary Table](#).

**Table 5-1160. EFUSE1\_ROW\_13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_13	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.39 EFUSE1\_ROW\_14 Register (Offset = 430h) [Reset = 00000000h]

EFUSE1\_ROW\_14 is shown in [Table 5-1161](#).

Return to the [Summary Table](#).

**Table 5-1161. EFUSE1\_ROW\_14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_14	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.40 EFUSE1\_ROW\_15 Register (Offset = 434h) [Reset = 00000000h]

EFUSE1\_ROW\_15 is shown in [Table 5-1162](#).

Return to the [Summary Table](#).

**Table 5-1162. EFUSE1\_ROW\_15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_15	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

#### 5.2.9.41 EFUSE1\_ROW\_16 Register (Offset = 438h) [Reset = 00000000h]

EFUSE1\_ROW\_16 is shown in [Table 5-1163](#).

Return to the [Summary Table](#).

**Table 5-1163. EFUSE1\_ROW\_16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_16	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

#### 5.2.9.42 EFUSE1\_ROW\_17 Register (Offset = 43Ch) [Reset = 00000000h]

EFUSE1\_ROW\_17 is shown in [Table 5-1164](#).

Return to the [Summary Table](#).

**Table 5-1164. EFUSE1\_ROW\_17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_17	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.43 EFUSE1\_ROW\_18 Register (Offset = 440h) [Reset = 00000000h]

EFUSE1\_ROW\_18 is shown in [Table 5-1165](#).

Return to the [Summary Table](#).

**Table 5-1165. EFUSE1\_ROW\_18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_18	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details



#### 5.2.9.44 EFUSE1\_ROW\_19 Register (Offset = 444h) [Reset = 00000000h]

EFUSE1\_ROW\_19 is shown in [Table 5-1166](#).

Return to the [Summary Table](#).

**Table 5-1166. EFUSE1\_ROW\_19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_19	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.45 EFUSE1\_ROW\_20 Register (Offset = 448h) [Reset = 00000000h]

EFUSE1\_ROW\_20 is shown in [Table 5-1167](#).

Return to the [Summary Table](#).

**Table 5-1167. EFUSE1\_ROW\_20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_20	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.46 EFUSE1\_ROW\_21 Register (Offset = 44Ch) [Reset = 00000000h]

EFUSE1\_ROW\_21 is shown in [Table 5-1168](#).

Return to the [Summary Table](#).

**Table 5-1168. EFUSE1\_ROW\_21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_21	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.47 EFUSE1\_ROW\_22 Register (Offset = 450h) [Reset = 00000000h]

EFUSE1\_ROW\_22 is shown in [Table 5-1169](#).

Return to the [Summary Table](#).

**Table 5-1169. EFUSE1\_ROW\_22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_22	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.48 EFUSE1\_ROW\_23 Register (Offset = 454h) [Reset = 00000000h]

EFUSE1\_ROW\_23 is shown in [Table 5-1170](#).

Return to the [Summary Table](#).

**Table 5-1170. EFUSE1\_ROW\_23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_23	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.49 EFUSE1\_ROW\_24 Register (Offset = 458h) [Reset = 00000000h]

EFUSE1\_ROW\_24 is shown in [Table 5-1171](#).

Return to the [Summary Table](#).

**Table 5-1171. EFUSE1\_ROW\_24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_24	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.50 EFUSE1\_ROW\_25 Register (Offset = 45Ch) [Reset = 00000000h]

EFUSE1\_ROW\_25 is shown in [Table 5-1172](#).

Return to the [Summary Table](#).

**Table 5-1172. EFUSE1\_ROW\_25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_25	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.51 EFUSE1\_ROW\_26 Register (Offset = 460h) [Reset = 00000000h]

EFUSE1\_ROW\_26 is shown in [Table 5-1173](#).

Return to the [Summary Table](#).

**Table 5-1173. EFUSE1\_ROW\_26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_26	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details



### 5.2.9.52 EFUSE1\_ROW\_27 Register (Offset = 464h) [Reset = 00000000h]

EFUSE1\_ROW\_27 is shown in [Table 5-1174](#).

Return to the [Summary Table](#).

**Table 5-1174. EFUSE1\_ROW\_27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_27	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.53 EFUSE1\_ROW\_28 Register (Offset = 468h) [Reset = 00000000h]

EFUSE1\_ROW\_28 is shown in [Table 5-1175](#).

Return to the [Summary Table](#).

**Table 5-1175. EFUSE1\_ROW\_28 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_28	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

#### 5.2.9.54 EFUSE1\_ROW\_29 Register (Offset = 46Ch) [Reset = 00000000h]

EFUSE1\_ROW\_29 is shown in [Table 5-1176](#).

Return to the [Summary Table](#).

**Table 5-1176. EFUSE1\_ROW\_29 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_29	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.55 EFUSE1\_ROW\_30 Register (Offset = 470h) [Reset = 00000000h]

EFUSE1\_ROW\_30 is shown in [Table 5-1177](#).

Return to the [Summary Table](#).

**Table 5-1177. EFUSE1\_ROW\_30 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_30	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.56 EFUSE1\_ROW\_31 Register (Offset = 474h) [Reset = 00000000h]

EFUSE1\_ROW\_31 is shown in [Table 5-1178](#).

Return to the [Summary Table](#).

**Table 5-1178. EFUSE1\_ROW\_31 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_31	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.57 EFUSE1\_ROW\_32 Register (Offset = 478h) [Reset = 00000000h]

EFUSE1\_ROW\_32 is shown in [Table 5-1179](#).

Return to the [Summary Table](#).

**Table 5-1179. EFUSE1\_ROW\_32 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_32	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.58 EFUSE1\_ROW\_33 Register (Offset = 47Ch) [Reset = 00000000h]

EFUSE1\_ROW\_33 is shown in [Table 5-1180](#).

Return to the [Summary Table](#).

**Table 5-1180. EFUSE1\_ROW\_33 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_33	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.59 EFUSE1\_ROW\_34 Register (Offset = 480h) [Reset = 00000000h]

EFUSE1\_ROW\_34 is shown in [Table 5-1181](#).

Return to the [Summary Table](#).

**Table 5-1181. EFUSE1\_ROW\_34 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_34	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details



### 5.2.9.60 EFUSE1\_ROW\_35 Register (Offset = 484h) [Reset = 00000000h]

EFUSE1\_ROW\_35 is shown in [Table 5-1182](#).

Return to the [Summary Table](#).

**Table 5-1182. EFUSE1\_ROW\_35 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_35	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.61 EFUSE1\_ROW\_36 Register (Offset = 488h) [Reset = 00000000h]

EFUSE1\_ROW\_36 is shown in [Table 5-1183](#).

Return to the [Summary Table](#).

**Table 5-1183. EFUSE1\_ROW\_36 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_36	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.62 EFUSE1\_ROW\_37 Register (Offset = 48Ch) [Reset = 00000000h]

EFUSE1\_ROW\_37 is shown in [Table 5-1184](#).

Return to the [Summary Table](#).

**Table 5-1184. EFUSE1\_ROW\_37 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_37	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.63 EFUSE1\_ROW\_38 Register (Offset = 490h) [Reset = 00000000h]

EFUSE1\_ROW\_38 is shown in [Table 5-1185](#).

Return to the [Summary Table](#).

**Table 5-1185. EFUSE1\_ROW\_38 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_38	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

#### 5.2.9.64 EFUSE1\_ROW\_39 Register (Offset = 494h) [Reset = 00000000h]

EFUSE1\_ROW\_39 is shown in [Table 5-1186](#).

Return to the [Summary Table](#).

**Table 5-1186. EFUSE1\_ROW\_39 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_39	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.65 EFUSE1\_ROW\_40 Register (Offset = 498h) [Reset = 00000000h]

EFUSE1\_ROW\_40 is shown in [Table 5-1187](#).

Return to the [Summary Table](#).

**Table 5-1187. EFUSE1\_ROW\_40 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_40	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.66 EFUSE1\_ROW\_41 Register (Offset = 49Ch) [Reset = 00000000h]

EFUSE1\_ROW\_41 is shown in [Table 5-1188](#).

Return to the [Summary Table](#).

**Table 5-1188. EFUSE1\_ROW\_41 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_41	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.67 EFUSE1\_ROW\_42 Register (Offset = 4A0h) [Reset = 00000000h]

EFUSE1\_ROW\_42 is shown in [Table 5-1189](#).

Return to the [Summary Table](#).

**Table 5-1189. EFUSE1\_ROW\_42 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_42	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details



### 5.2.9.68 EFUSE1\_ROW\_43 Register (Offset = 4A4h) [Reset = 00000000h]

EFUSE1\_ROW\_43 is shown in [Table 5-1190](#).

Return to the [Summary Table](#).

**Table 5-1190. EFUSE1\_ROW\_43 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-0	EFUSE1_ROW_43	R	0h	Captures the EFUSE Value. Refer to EFUSE Mapping XLS for more details

### 5.2.9.69 EFUSE\_OVERRIDE\_HSM\_HALT\_ON\_ROM\_ECC\_ERR\_EN Register (Offset = 800h) [Reset = 0000000Xh]

EFUSE\_OVERRIDE\_HSM\_HALT\_ON\_ROM\_ECC\_ERR\_EN is shown in [Table 5-1191](#).

Return to the [Summary Table](#).

**Table 5-1191. EFUSE\_OVERRIDE\_HSM\_HALT\_ON\_ROM\_ECC\_ERR\_EN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	override_val	R/W	0h	Override MMR value
3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value

### 5.2.9.70 EFUSE\_OVERRIDE\_MEM\_MARGINCTRL Register (Offset = 804h) [Reset = 0XXX0XXXh]

EFUSE\_OVERRIDE\_MEM\_MARGINCTRL is shown in [Table 5-1192](#).

Return to the [Summary Table](#).

**Table 5-1192. EFUSE\_OVERRIDE\_MEM\_MARGINCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	0h	
29-28	brg_margin	R/W	0h	Override MMR value
27	RESERVED	R/W	0h	
26-24	brg_margin_override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value
23-22	RESERVED	R/W	0h	
21-20	byg_margin	R/W	0h	Override MMR value
19	RESERVED	R/W	0h	
18-16	byg_margin_override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value
15-12	gwg_margin	R/W	0h	Override MMR value
11	RESERVED	R/W	0h	
10-8	gwg_margin_override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value
7-6	RESERVED	R/W	0h	
5-4	glg_margin	R/W	0h	Override MMR value
3	RESERVED	R/W	0h	
2-0	glg_margin_override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value

### 5.2.9.71 EFUSE\_OVERRIDE\_LVDS\_BGAP\_TRIM Register (Offset = 808h) [Reset = 0000000Xh]

EFUSE\_OVERRIDE\_LVDS\_BGAP\_TRIM is shown in [Table 5-1193](#).

Return to the [Summary Table](#).

**Table 5-1193. EFUSE\_OVERRIDE\_LVDS\_BGAP\_TRIM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	0h	
9-4	override_val	R/W	0h	Override MMR value
3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value

### 5.2.9.72 EFUSE\_OVERRIDE\_XTAL\_STABLIZATION\_WAIT Register (Offset = 80Ch) [Reset = 00000000h]

EFUSE\_OVERRIDE\_XTAL\_STABLIZATION\_WAIT is shown in [Table 5-1194](#).

Return to the [Summary Table](#).

**Table 5-1194. EFUSE\_OVERRIDE\_XTAL\_STABLIZATION\_WAIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value . Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.73 EFUSE\_OVERRIDE\_SLICER\_BIAS\_RTRIM Register (Offset = 810h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_SLICER\_BIAS\_RTRIM is shown in [Table 5-1195](#).

Return to the [Summary Table](#).

**Table 5-1195. EFUSE\_OVERRIDE\_SLICER\_BIAS\_RTRIM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value . Refer to the ANAREG in TOP_RCM for the override value

#### 5.2.9.74 EFUSE\_OVERRIDE\_XO\_OUTPUT\_DRIVE Register (Offset = 814h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_XO\_OUTPUT\_DRIVE is shown in [Table 5-1196](#).

Return to the [Summary Table](#).

**Table 5-1196. EFUSE\_OVERRIDE\_XO\_OUTPUT\_DRIVE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value . Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.75 EFUSE\_OVERRIDE\_RCOSC\_TRIM\_CODE Register (Offset = 818h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_RCOSC\_TRIM\_CODE is shown in [Table 5-1197](#).

Return to the [Summary Table](#).

**Table 5-1197. EFUSE\_OVERRIDE\_RCOSC\_TRIM\_CODE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value . Refer to the ANAREG in TOP_RCM for the override value



### 5.2.9.76 EFUSE\_OVERRIDE\_IP1\_BG1\_RTRIM Register (Offset = 81Ch) [Reset = 00000000h]

EFUSE\_OVERRIDE\_IP1\_BG1\_RTRIM is shown in [Table 5-1198](#).

Return to the [Summary Table](#).

**Table 5-1198. EFUSE\_OVERRIDE\_IP1\_BG1\_RTRIM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.77 EFUSE\_OVERRIDE\_IP1\_BG1\_SLOPE Register (Offset = 820h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_IP1\_BG1\_SLOPE is shown in [Table 5-1199](#).

Return to the [Summary Table](#).

**Table 5-1199. EFUSE\_OVERRIDE\_IP1\_BG1\_SLOPE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.78 EFUSE\_OVERRIDE\_IP1\_BG1\_MAG Register (Offset = 824h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_IP1\_BG1\_MAG is shown in [Table 5-1200](#).

Return to the [Summary Table](#).

**Table 5-1200. EFUSE\_OVERRIDE\_IP1\_BG1\_MAG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.79 EFUSE\_OVERRIDE\_RS232\_CLKMODE Register (Offset = 828h) [Reset = 0000000Xh]

EFUSE\_OVERRIDE\_RS232\_CLKMODE is shown in [Table 5-1201](#).

Return to the [Summary Table](#).

**Table 5-1201. EFUSE\_OVERRIDE\_RS232\_CLKMODE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	override_val	R/W	0h	Override value for RS232 Clock Mode 0 : Autobaud 1 : Fixed Interval
3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value.

### 5.2.9.80 EFUSE\_OVERRIDE\_VMON\_VDD\_OV\_UV\_TRIM Register (Offset = 82Ch) [Reset = 00000000h]

EFUSE\_OVERRIDE\_VMON\_VDD\_OV\_UV\_TRIM is shown in [Table 5-1202](#).

Return to the [Summary Table](#).

**Table 5-1202. EFUSE\_OVERRIDE\_VMON\_VDD\_OV\_UV\_TRIM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.81 EFUSE\_OVERRIDE\_VMON\_VDDS\_3P3\_UV\_TRIM Register (Offset = 830h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_VMON\_VDDS\_3P3\_UV\_TRIM is shown in [Table 5-1203](#).

Return to the [Summary Table](#).

**Table 5-1203. EFUSE\_OVERRIDE\_VMON\_VDDS\_3P3\_UV\_TRIM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.82 EFUSE\_OVERRIDE\_VMON\_VDDA\_OSC\_TRIM Register (Offset = 834h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_VMON\_VDDA\_OSC\_TRIM is shown in [Table 5-1204](#).

Return to the [Summary Table](#).

**Table 5-1204. EFUSE\_OVERRIDE\_VMON\_VDDA\_OSC\_TRIM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.83 EFUSE\_OVERRIDE\_VDD\_VT\_DET Register (Offset = 838h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_VDD\_VT\_DET is shown in [Table 5-1205](#).

Return to the [Summary Table](#).

**Table 5-1205. EFUSE\_OVERRIDE\_VDD\_VT\_DET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value



#### 5.2.9.84 EFUSE\_OVERRIDE\_MASK\_CPU\_CLK\_OUT\_CTRL\_LOWV\_VAL Register (Offset = 83Ch) [Reset = 00000000h]

EFUSE\_OVERRIDE\_MASK\_CPU\_CLK\_OUT\_CTRL\_LOWV\_VAL is shown in [Table 5-1206](#).

Return to the [Summary Table](#).

**Table 5-1206. EFUSE\_OVERRIDE\_MASK\_CPU\_CLK\_OUT\_CTRL\_LOWV\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.85 EFUSE\_OVERRIDE\_MASK\_CPU\_CLK\_OUT\_CTRL\_LOWV\_SEL Register (Offset = 840h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_MASK\_CPU\_CLK\_OUT\_CTRL\_LOWV\_SEL is shown in [Table 5-1207](#).

Return to the [Summary Table](#).

**Table 5-1207. EFUSE\_OVERRIDE\_MASK\_CPU\_CLK\_OUT\_CTRL\_LOWV\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.86 EFUSE\_OVERRIDE\_EN\_VOL\_MON\_FUNC Register (Offset = 844h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_EN\_VOL\_MON\_FUNC is shown in [Table 5-1208](#).

Return to the [Summary Table](#).

**Table 5-1208. EFUSE\_OVERRIDE\_EN\_VOL\_MON\_FUNC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.87 EFUSE\_OVERRIDE\_SPARE\_ANA Register (Offset = 84Ch) [Reset = 00000000h]

EFUSE\_OVERRIDE\_SPARE\_ANA is shown in [Table 5-1209](#).

Return to the [Summary Table](#).

**Table 5-1209. EFUSE\_OVERRIDE\_SPARE\_ANA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.88 EFUSE\_OVERRIDE\_SLICER\_DLY\_DISABLE Register (Offset = 850h) [Reset = 00000000h]

EFUSE\_OVERRIDE\_SLICER\_DLY\_DISABLE is shown in [Table 5-1210](#).

Return to the [Summary Table](#).

**Table 5-1210. EFUSE\_OVERRIDE\_SLICER\_DLY\_DISABLE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	0h	
2-0	override	R/W	0h	Override EFUSE Value with SW Value Write 3'b000 : EFUSE Value Write 3'b111 : MMR Value. Refer to the ANAREG in TOP_RCM for the override value

### 5.2.9.89 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-1211](#).

Return to the [Summary Table](#).

**Table 5-1211. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.9.90 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-1212](#).

Return to the [Summary Table](#).

**Table 5-1212. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.9.91 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-1213](#).

Return to the [Summary Table](#).

**Table 5-1213. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D



### 5.2.9.92 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-1214](#).

Return to the [Summary Table](#).

**Table 5-1214. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.9.93 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-1215](#).

Return to the [Summary Table](#).

**Table 5-1215. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.9.94 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-1216](#).

Return to the [Summary Table](#).

**Table 5-1216. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

### 5.2.9.95 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-1217](#).

Return to the [Summary Table](#).

**Table 5-1217. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D

### 5.2.9.96 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-1218](#).

Return to the [Summary Table](#).

**Table 5-1218. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.9.97 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-1219](#).

Return to the [Summary Table](#).

**Table 5-1219. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

### 5.2.9.98 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-1220](#).

Return to the [Summary Table](#).

**Table 5-1220. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.9.99 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-1221](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-1221. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component



### 5.2.9.100 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-1222](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-1222. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.9.101 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-1223](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-1223. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.9.102 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-1224](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-1224. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.9.103 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-1225](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-1225. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

### 5.2.9.104 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-1226](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-1226. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.9.105 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-1227](#).

Return to the [Summary Table](#).

EOI register

**Table 5-1227. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.9.106 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-1228](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-1228. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.9.107 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-1229](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-1229. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault



### 5.2.9.108 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-1230](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-1230. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.9.109 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-1231](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-1231. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

### 5.2.10 RSS\_PROC\_CTRL Registers

Table 5-1232 lists the memory-mapped registers for the RSS\_PROC\_CTRL registers. All register offset addresses not listed in Table 5-1232 should be considered as reserved locations and the register contents should not be modified.

**Table 5-1232. RSS\_PROC\_CTRL Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID register	<a href="#">Section 5.2.10.1</a>
4h	HW_REG0		<a href="#">Section 5.2.10.2</a>
8h	RSS_CR4_MBOX_WRITE_DONE		<a href="#">Section 5.2.10.3</a>
Ch	RSS_CR4_MBOX_READ_REQ		<a href="#">Section 5.2.10.4</a>
10h	RSS_CR4_MBOX_READ_DONE		<a href="#">Section 5.2.10.5</a>
14h	RSS_CR4_MBOX_READ_DONE_ACK		<a href="#">Section 5.2.10.6</a>
18h	BSS_TO_SOC_SW_INT		<a href="#">Section 5.2.10.7</a>
1Ch	RSS_TO_MSS_INT_MASK		<a href="#">Section 5.2.10.8</a>
20h	RSS_CR4_BOOT_INFO_REG0		<a href="#">Section 5.2.10.9</a>
24h	RSS_CR4_BOOT_INFO_REG1		<a href="#">Section 5.2.10.10</a>
28h	RSS_CR4_BOOT_INFO_REG2		<a href="#">Section 5.2.10.11</a>
2Ch	RSS_CR4_BOOT_INFO_REG3		<a href="#">Section 5.2.10.12</a>
30h	RSS_CR4_BOOT_INFO_REG4		<a href="#">Section 5.2.10.13</a>
34h	RSS_CR4_BOOT_INFO_REG5		<a href="#">Section 5.2.10.14</a>
38h	RSS_CR4_BOOT_INFO_REG6		<a href="#">Section 5.2.10.15</a>
3Ch	RSS_CR4_BOOT_INFO_REG7		<a href="#">Section 5.2.10.16</a>
40h	RSS_STATIC_MEM_MEMINIT		<a href="#">Section 5.2.10.17</a>
44h	RSS_STATIC_MEM_MEMINIT_DONE		<a href="#">Section 5.2.10.18</a>
48h	RSS_STATIC_MEM_MEMINIT_STATUS		<a href="#">Section 5.2.10.19</a>
4Ch	RSS_MBOX_MEMINIT		<a href="#">Section 5.2.10.20</a>
50h	RSS_MBOX_MEMINIT_DONE		<a href="#">Section 5.2.10.21</a>
54h	RSS_MBOX_MEMINIT_STATUS		<a href="#">Section 5.2.10.22</a>
400h	FRCCFG		<a href="#">Section 5.2.10.23</a>
404h	MSSEVNTGEN1LSB		<a href="#">Section 5.2.10.24</a>
408h	MSSEVNTGEN1MID		<a href="#">Section 5.2.10.25</a>
40Ch	MSSEVNTGEN1MSB		<a href="#">Section 5.2.10.26</a>
410h	MSSEVNTGEN2LSB		<a href="#">Section 5.2.10.27</a>
414h	MSSEVNTGEN2MID		<a href="#">Section 5.2.10.28</a>
418h	MSSEVNTGEN2MSB		<a href="#">Section 5.2.10.29</a>
41Ch	MSSEVNTGEN3LSB		<a href="#">Section 5.2.10.30</a>
420h	MSSEVNTGEN3MID		<a href="#">Section 5.2.10.31</a>
424h	MSSEVNTGEN3MSB		<a href="#">Section 5.2.10.32</a>
428h	MSSEVNTGEN4LSB		<a href="#">Section 5.2.10.33</a>
42Ch	MSSEVNTGEN4MID		<a href="#">Section 5.2.10.34</a>
430h	MSSEVNTGEN4MSB		<a href="#">Section 5.2.10.35</a>
434h	TIMSTMP1LSB		<a href="#">Section 5.2.10.36</a>
438h	TIMSTMP1MID		<a href="#">Section 5.2.10.37</a>
43Ch	TIMSTMP1MSB		<a href="#">Section 5.2.10.38</a>
440h	TIMSTMP2LSB		<a href="#">Section 5.2.10.39</a>
444h	TIMSTMP2MID		<a href="#">Section 5.2.10.40</a>

**Table 5-1232. RSS\_PROC\_CTRL Registers (continued)**

Offset	Acronym	Register Name	Section
448h	TIMSTMP2MSB		<a href="#">Section 5.2.10.41</a>
44Ch	TIMSTMP3LSB		<a href="#">Section 5.2.10.42</a>
450h	TIMSTMP3MID		<a href="#">Section 5.2.10.43</a>
454h	TIMSTMP3MSB		<a href="#">Section 5.2.10.44</a>
458h	TIMSTMP4LSB		<a href="#">Section 5.2.10.45</a>
45Ch	TIMSTMP4MID		<a href="#">Section 5.2.10.46</a>
460h	TIMSTMP4MSB		<a href="#">Section 5.2.10.47</a>
464h	FRCLSBCNT		<a href="#">Section 5.2.10.48</a>
468h	FRCMIDCNT		<a href="#">Section 5.2.10.49</a>
46Ch	FRCMSBCNT		<a href="#">Section 5.2.10.50</a>
470h	TIMSTMPTRIGSEL		<a href="#">Section 5.2.10.51</a>
474h	FRCLSBPERIOD		<a href="#">Section 5.2.10.52</a>
478h	FRCMIDPERIOD		<a href="#">Section 5.2.10.53</a>
47Ch	FRCLSBCNTINITVAL		<a href="#">Section 5.2.10.54</a>
480h	FRCMIDCNTINITVAL		<a href="#">Section 5.2.10.55</a>
484h	FRCMSBCNTINITVAL		<a href="#">Section 5.2.10.56</a>
488h	FRCSYNCOUTLSBCNT		<a href="#">Section 5.2.10.57</a>
48Ch	FRCSYNCOUTMIDCNT		<a href="#">Section 5.2.10.58</a>
490h	FRCRESETMODE		<a href="#">Section 5.2.10.59</a>
494h	FRCCFG2		<a href="#">Section 5.2.10.60</a>
FD0h	HW_SPARE_RW0		<a href="#">Section 5.2.10.61</a>
FD4h	HW_SPARE_RW1		<a href="#">Section 5.2.10.62</a>
FD8h	HW_SPARE_RW2		<a href="#">Section 5.2.10.63</a>
FDCh	HW_SPARE_RW3		<a href="#">Section 5.2.10.64</a>
FE0h	HW_SPARE_RO0		<a href="#">Section 5.2.10.65</a>
FE4h	HW_SPARE_RO1		<a href="#">Section 5.2.10.66</a>
FE8h	HW_SPARE_RO2		<a href="#">Section 5.2.10.67</a>
FECh	HW_SPARE_RO3		<a href="#">Section 5.2.10.68</a>
FF0h	HW_SPARE_WPH		<a href="#">Section 5.2.10.69</a>
FF4h	HW_SPARE_REC		<a href="#">Section 5.2.10.70</a>
1008h	LOCK0_KICK0	- KICK0 component	<a href="#">Section 5.2.10.71</a>
100Ch	LOCK0_KICK1	- KICK1 component	<a href="#">Section 5.2.10.72</a>
1010h	intr_raw_status	Interrupt Raw Status/Set Register	<a href="#">Section 5.2.10.73</a>
1014h	intr_enabled_status_clear	Interrupt Enabled Status/Clear register	<a href="#">Section 5.2.10.74</a>
1018h	intr_enable	Interrupt Enable register	<a href="#">Section 5.2.10.75</a>
101Ch	intr_enable_clear	Interrupt Enable Clear register	<a href="#">Section 5.2.10.76</a>
1020h	eoi	EOI register	<a href="#">Section 5.2.10.77</a>
1024h	fault_address	Fault Address register	<a href="#">Section 5.2.10.78</a>
1028h	fault_type_status	Fault Type Status register	<a href="#">Section 5.2.10.79</a>
102Ch	fault_attr_status	Fault Attribute Status register	<a href="#">Section 5.2.10.80</a>
1030h	fault_clear	Fault Clear register	<a href="#">Section 5.2.10.81</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-1233](#) shows the codes that are used for access types in this section.

**Table 5-1233. RSS\_PROC\_CTRL Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.10.1 PID Register (Offset = 0h) [Reset = 61800214h]

PID is shown in [Table 5-1234](#).

Return to the [Summary Table](#).

PID register

**Table 5-1234. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	PID_msb16	R	6180h	
15-11	PID_misc	R	0h	
10-8	PID_major	R	2h	
7-6	PID_custom	R	0h	
5-0	PID_minor	R	14h	

### 5.2.10.2 HW\_REG0 Register (Offset = 4h) [Reset = 00000000h]

HW\_REG0 is shown in [Table 5-1235](#).

Return to the [Summary Table](#).

**Table 5-1235. HW\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	reg	R/W	0h	TI Reserved

### 5.2.10.3 RSS\_CR4\_MBOX\_WRITE\_DONE Register (Offset = 8h) [Reset = 0XXXXXXh]

RSS\_CR4\_MBOX\_WRITE\_DONE is shown in [Table 5-1236](#).

Return to the [Summary Table](#).

**Table 5-1236. RSS\_CR4\_MBOX\_WRITE\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	proc_7	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 7
27-25	RESERVED	R/W	0h	
24	proc_6	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 6
23-21	RESERVED	R/W	0h	
20	proc_5	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 5
19-17	RESERVED	R/W	0h	
16	proc_4	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 4
15-13	RESERVED	R/W	0h	
12	proc_3	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 3
11-9	RESERVED	R/W	0h	
8	proc_2	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 2
7-5	RESERVED	R/W	0h	
4	proc_1	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 1
3-1	RESERVED	R/W	0h	
0	proc_0	R/W	0h	This register should be written once finishing writing into the mailbox memory of processor 0



#### 5.2.10.4 RSS\_CR4\_MBOX\_READ\_REQ Register (Offset = Ch) [Reset = 0XXXXXXXh]

RSS\_CR4\_MBOX\_READ\_REQ is shown in [Table 5-1237](#).

Return to the [Summary Table](#).

**Table 5-1237. RSS\_CR4\_MBOX\_READ\_REQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	proc_7	R/W	0h	This is request from processor 7 to mss_cr5a. Requesting it to read from mailbox.
27-25	RESERVED	R/W	0h	
24	proc_6	R/W	0h	This is request from processor 6 to mss_cr5a. Requesting it to read from mailbox.
23-21	RESERVED	R/W	0h	
20	proc_5	R/W	0h	This is request from processor 5 to mss_cr5a. Requesting it to read from mailbox.
19-17	RESERVED	R/W	0h	
16	proc_4	R/W	0h	This is request from processor 4 to mss_cr5a. Requesting it to read from mailbox.
15-13	RESERVED	R/W	0h	
12	proc_3	R/W	0h	This is request from processor 3 to mss_cr5a. Requesting it to read from mailbox.
11-9	RESERVED	R/W	0h	
8	proc_2	R/W	0h	This is request from processor 2 to mss_cr5a. Requesting it to read from mailbox.
7-5	RESERVED	R/W	0h	
4	proc_1	R/W	0h	This is request from processor 1 to mss_cr5a. Requesting it to read from mailbox.
3-1	RESERVED	R/W	0h	
0	proc_0	R/W	0h	This is request from processor 0 to mss_cr5a. Requesting it to read from mailbox.

### 5.2.10.5 RSS\_CR4\_MBOX\_READ\_DONE Register (Offset = 10h) [Reset = 0XXXXXXXh]

RSS\_CR4\_MBOX\_READ\_DONE is shown in [Table 5-1238](#).

Return to the [Summary Table](#).

**Table 5-1238. RSS\_CR4\_MBOX\_READ\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28	proc_7	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 7
27-25	RESERVED	R/W	0h	
24	proc_6	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 6
23-21	RESERVED	R/W	0h	
20	proc_5	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 5
19-17	RESERVED	R/W	0h	
16	proc_4	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 4
15-13	RESERVED	R/W	0h	
12	proc_3	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 3
11-9	RESERVED	R/W	0h	
8	proc_2	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 2
7-5	RESERVED	R/W	0h	
4	proc_1	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 1
3-1	RESERVED	R/W	0h	
0	proc_0	R/W	0h	This register should be written once finishing reading from CR5A's mailbox written by proc 0

### 5.2.10.6 RSS\_CR4\_MBOX\_READ\_DONE\_ACK Register (Offset = 14h) [Reset = 00000000h]

RSS\_CR4\_MBOX\_READ\_DONE\_ACK is shown in [Table 5-1239](#).

Return to the [Summary Table](#).

**Table 5-1239. RSS\_CR4\_MBOX\_READ\_DONE\_ACK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	proc	R/W	0h	Write pulse bit field: For bits 0 to 7: Wrting 1'b1 : Generates pulse interrupt to corresponding proc from RSS_CR4

### 5.2.10.7 BSS\_TO\_SOC\_SW\_INT Register (Offset = 18h) [Reset = 00000000h]

BSS\_TO\_SOC\_SW\_INT is shown in [Table 5-1240](#).

Return to the [Summary Table](#).

**Table 5-1240. BSS\_TO\_SOC\_SW\_INT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3-0	trig	R/W	0h	Write pulse bit field: For bits 0 to 3: Wrting 1'b1 : Generates DSS_SW_INT_RSS_PROC_CTRL<0-3>

### 5.2.10.8 RSS\_TO\_MSS\_INT\_MASK Register (Offset = 1Ch) [Reset = 0000001Xh]

RSS\_TO\_MSS\_INT\_MASK is shown in [Table 5-1241](#).

Return to the [Summary Table](#).

**Table 5-1241. RSS\_TO\_MSS\_INT\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R/W	0h	
4	ESM_LO	R/W	1h	writing: 1'b1: Mask ESM-LO interrupt reaching DSS and MSS 1'b0: Unmasks ESM-LO interrupt reaching DSS and MSS
3-1	RESERVED	R/W	0h	
0	ESM_HI	R/W	1h	writing: 1'b1: Mask ESM-HI interrupt reaching DSS and MSS 1'b0: Unmasks ESM-HI interrupt reaching DSS and MSS

### 5.2.10.9 RSS\_CR4\_BOOT\_INFO\_REG0 Register (Offset = 20h) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG0 is shown in [Table 5-1242](#).

Return to the [Summary Table](#).

**Table 5-1242. RSS\_CR4\_BOOT\_INFO\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

#### 5.2.10.10 RSS\_CR4\_BOOT\_INFO\_REG1 Register (Offset = 24h) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG1 is shown in [Table 5-1243](#).

Return to the [Summary Table](#).

**Table 5-1243. RSS\_CR4\_BOOT\_INFO\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.10.11 RSS\_CR4\_BOOT\_INFO\_REG2 Register (Offset = 28h) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG2 is shown in [Table 5-1244](#).

Return to the [Summary Table](#).

**Table 5-1244. RSS\_CR4\_BOOT\_INFO\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use



### 5.2.10.12 RSS\_CR4\_BOOT\_INFO\_REG3 Register (Offset = 2Ch) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG3 is shown in [Table 5-1245](#).

Return to the [Summary Table](#).

**Table 5-1245. RSS\_CR4\_BOOT\_INFO\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.10.13 RSS\_CR4\_BOOT\_INFO\_REG4 Register (Offset = 30h) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG4 is shown in [Table 5-1246](#).

Return to the [Summary Table](#).

**Table 5-1246. RSS\_CR4\_BOOT\_INFO\_REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

#### 5.2.10.14 RSS\_CR4\_BOOT\_INFO\_REG5 Register (Offset = 34h) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG5 is shown in [Table 5-1247](#).

Return to the [Summary Table](#).

**Table 5-1247. RSS\_CR4\_BOOT\_INFO\_REG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.10.15 RSS\_CR4\_BOOT\_INFO\_REG6 Register (Offset = 38h) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG6 is shown in [Table 5-1248](#).

Return to the [Summary Table](#).

**Table 5-1248. RSS\_CR4\_BOOT\_INFO\_REG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.10.16 RSS\_CR4\_BOOT\_INFO\_REG7 Register (Offset = 3Ch) [Reset = 00000000h]

RSS\_CR4\_BOOT\_INFO\_REG7 is shown in [Table 5-1249](#).

Return to the [Summary Table](#).

**Table 5-1249. RSS\_CR4\_BOOT\_INFO\_REG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	config	R/W	0h	Reserved Register for Software use

### 5.2.10.17 RSS\_STATIC\_MEM\_MEMINIT Register (Offset = 40h) [Reset = 00000000h]

RSS\_STATIC\_MEM\_MEMINIT is shown in [Table 5-1250](#).

Return to the [Summary Table](#).

**Table 5-1250. RSS\_STATIC\_MEM\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	start	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.

### 5.2.10.18 RSS\_STATIC\_MEM\_MEMINIT\_DONE Register (Offset = 44h) [Reset = 00000000h]

RSS\_STATIC\_MEM\_MEMINIT\_DONE is shown in [Table 5-1251](#).

Return to the [Summary Table](#).

**Table 5-1251. RSS\_STATIC\_MEM\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.10.19 RSS\_STATIC\_MEM\_MEMINIT\_STATUS Register (Offset = 48h) [Reset = 00000000h]

RSS\_STATIC\_MEM\_MEMINIT\_STATUS is shown in [Table 5-1252](#).

Return to the [Summary Table](#).

**Table 5-1252. RSS\_STATIC\_MEM\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.



### 5.2.10.20 RSS\_MBOX\_MEMINIT Register (Offset = 4Ch) [Reset = 00000000h]

RSS\_MBOX\_MEMINIT is shown in [Table 5-1253](#).

Return to the [Summary Table](#).

**Table 5-1253. RSS\_MBOX\_MEMINIT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	start	R/W	0h	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.

### 5.2.10.21 RSS\_MBOX\_MEMINIT\_DONE Register (Offset = 50h) [Reset = 00000000h]

RSS\_MBOX\_MEMINIT\_DONE is shown in [Table 5-1254](#).

Return to the [Summary Table](#).

**Table 5-1254. RSS\_MBOX\_MEMINIT\_DONE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	
0	done	R/W	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.

### 5.2.10.22 RSS\_MBOX\_MEMINIT\_STATUS Register (Offset = 54h) [Reset = 00000000h]

RSS\_MBOX\_MEMINIT\_STATUS is shown in [Table 5-1255](#).

Return to the [Summary Table](#).

**Table 5-1255. RSS\_MBOX\_MEMINIT\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	status	R	0h	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.

### 5.2.10.23 FRCCFG Register (Offset = 400h) [Reset = 0007FX0Xh]

FRCCFG is shown in [Table 5-1256](#).

Return to the [Summary Table](#).

**Table 5-1256. FRCCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R/W	0h	
28-26	FRCSYNCINSRCSEL	R/W	0h	FRC Sync In Source Select. 0-->SW Sync In, 1-->External Sync In
25	FRCCFGSWRST	R/W	0h	FRC Software reset. Self clearing Pulse.
24	FRCCMPDIS	R/W	0h	FRC Parity Compare Disable
23	FRCCMPXORININV	R/W	0h	FRC Parity Compare Logic input XOR Inversion.
22	FRCCMPXOROUTINV	R/W	0h	FRC Parity Compare Logic output XOR Inversion.
21	FRCCMPOROUTMSK	R/W	0h	Parity Compare Logic OR Out Mask
20	FRCCMPANDOUTMSK	R/W	0h	Parity Compare Logic AND Out Mask
19	FRCSSYNCIN	R/W	0h	FRC Software Sync In signal.
18-17	MSSEVENTGEN1UMOD	R/W	3h	MSS Event Gen 1 Usage Mode. 00-->Only LSB, 01-->LSB+MID, 10,11-->All 3 Part.
16-15	MSSEVENTGEN2UMOD	R/W	3h	MSS Event Gen 2 Usage Mode. 00-->Only LSB, 01-->LSB+MID, 10,11-->All 3 Part.
14-13	MSSEVENTGEN3UMOD	R/W	3h	MSS Event Gen 3 Usage Mode. 00-->Only LSB, 01-->LSB+MID, 10,11-->All 3 Part.
12-11	MSSEVENTGEN4UMOD	R/W	3h	MSS Event Gen 4 Usage Mode. 00-->Only LSB, 01-->LSB+MID, 10,11-->All 3 Part.
10	MSSSWTRIG1	R/W	0h	MSS Software Trigger 1.. Self clearing Pulse
9	MSSSWTRIG2	R/W	0h	MSS Software Trigger 2. Self clearing Pulse
8	RESERVED	R/W	0h	
7	FRCEN	R/W	0h	Enable for FRC
6	FRCADCCLKENGATE	R/W	0h	Gating signal for ADC clock enable pulse sent to analog. 0 --> Allow the signal to pass through. 1--> Gate off FRC Sync Out signal.
5	FRCSYNCGATE	R/W	0h	Gating signal for ext_sync_in signal. 0 --> Allow the signal to pass through. 1--> Gate off FRC Sync Out signal.
4	FRCSYNCGOUTGATE	R/W	0h	Gating signal for ext_sync_out pulse. 0 --> Allow the signal to pass through. 1--> Gate off FRC Sync Out signal.
3	FRCSYNCGOUTMODE	R/W	0h	0 Uses only LSB counter for frame pulse generation. 1 Uses both LSB and MID counters for frame pulse generation.
2-1	FRCADCCLKENSEL	R/W	0h	ADC Clock Enable Select. 00,11-->Ext Sync In, 01-->LSB Roll-over, 10-->MID Counter Rollover
0	RESERVED	R/W	0h	

#### 5.2.10.24 MSSEVNTGEN1LSB Register (Offset = 404h) [Reset = 0FFFFFFFh]

MSSEVNTGEN1LSB is shown in [Table 5-1257](#).

Return to the [Summary Table](#).

**Table 5-1257. MSSEVNTGEN1LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN1LSB	R/W	0FFFFFFFh	MSS Event Gen 1 LSB Count value.

### 5.2.10.25 MSSEVNTGEN1MID Register (Offset = 408h) [Reset = 00000FFFh]

MSSEVNTGEN1MID is shown in [Table 5-1258](#).

Return to the [Summary Table](#).

**Table 5-1258. MSSEVNTGEN1MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	MSSEVNTGEN1MID	R/W	FFFh	MSS Event Gen 1 MID Count value.

### 5.2.10.26 MSSEVNTGEN1MSB Register (Offset = 40Ch) [Reset = 0FFFFFFFh]

MSSEVNTGEN1MSB is shown in [Table 5-1259](#).

Return to the [Summary Table](#).

**Table 5-1259. MSSEVNTGEN1MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN1MSB	R/W	0FFFFFFFh	MSS Event Gen 1 MSB Count value.

### 5.2.10.27 MSSEVNTGEN2LSB Register (Offset = 410h) [Reset = 0FFFFFFFh]

MSSEVNTGEN2LSB is shown in [Table 5-1260](#).

Return to the [Summary Table](#).

**Table 5-1260. MSSEVNTGEN2LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN2LSB	R/W	0FFFFFFFh	MSS Event Gen 2 LSB Count value.



### 5.2.10.28 MSSEVNTGEN2MID Register (Offset = 414h) [Reset = 00000FFFh]

MSSEVNTGEN2MID is shown in [Table 5-1261](#).

Return to the [Summary Table](#).

**Table 5-1261. MSSEVNTGEN2MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	MSSEVNTGEN2MID	R/W	FFFh	MSS Event Gen 2 MID Count value.

### 5.2.10.29 MSSEVNTGEN2MSB Register (Offset = 418h) [Reset = 0FFFFFFFh]

MSSEVNTGEN2MSB is shown in [Table 5-1262](#).

Return to the [Summary Table](#).

**Table 5-1262. MSSEVNTGEN2MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN2MSB	R/W	0FFFFFFFh	MSS Event Gen 2 MSB Count value.

### 5.2.10.30 MSSEVNTGEN3LSB Register (Offset = 41Ch) [Reset = 0FFFFFFFh]

MSSEVNTGEN3LSB is shown in [Table 5-1263](#).

Return to the [Summary Table](#).

**Table 5-1263. MSSEVNTGEN3LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN3LSB	R/W	0FFFFFFFh	MSS Event Gen 3 LSB Count value.

### 5.2.10.31 MSSEVNTGEN3MID Register (Offset = 420h) [Reset = 00000FFFh]

MSSEVNTGEN3MID is shown in [Table 5-1264](#).

Return to the [Summary Table](#).

**Table 5-1264. MSSEVNTGEN3MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	MSSEVNTGEN3MID	R/W	FFFh	MSS Event Gen 3 MID Count value.

### 5.2.10.32 MSSEVNTGEN3MSB Register (Offset = 424h) [Reset = 0FFFFFFFh]

MSSEVNTGEN3MSB is shown in [Table 5-1265](#).

Return to the [Summary Table](#).

**Table 5-1265. MSSEVNTGEN3MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN3MSB	R/W	0FFFFFFFh	MSS Event Gen 3 MSB Count value.

### 5.2.10.33 MSSEVNTGEN4LSB Register (Offset = 428h) [Reset = 0FFFFFFFh]

MSSEVNTGEN4LSB is shown in [Table 5-1266](#).

Return to the [Summary Table](#).

**Table 5-1266. MSSEVNTGEN4LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN4LSB	R/W	0FFFFFFFh	MSS Event Gen 4 LSB Count value.

### 5.2.10.34 MSSEVNTGEN4MID Register (Offset = 42Ch) [Reset = 0000FFFh]

MSSEVNTGEN4MID is shown in [Table 5-1267](#).

Return to the [Summary Table](#).

**Table 5-1267. MSSEVNTGEN4MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	MSSEVNTGEN4MID	R/W	FFFh	MSS Event Gen 4 MID Count value.

### 5.2.10.35 MSSEVNTGEN4MSB Register (Offset = 430h) [Reset = 0FFFFFFFh]

MSSEVNTGEN4MSB is shown in [Table 5-1268](#).

Return to the [Summary Table](#).

**Table 5-1268. MSSEVNTGEN4MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	MSSEVNTGEN4MSB	R/W	0FFFFFFFh	MSS Event Gen 4 MSB Count value.



### 5.2.10.36 TIMSTMP1LSB Register (Offset = 434h) [Reset = 00000000h]

TIMSTMP1LSB is shown in [Table 5-1269](#).

Return to the [Summary Table](#).

**Table 5-1269. TIMSTMP1LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP1LSB	R	0h	Time Stamp 1 LSB Count Output

### 5.2.10.37 TIMSTMP1MID Register (Offset = 438h) [Reset = 00000000h]

TIMSTMP1MID is shown in [Table 5-1270](#).

Return to the [Summary Table](#).

**Table 5-1270. TIMSTMP1MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP1MID	R	0h	Time Stamp 1 MID Count Output

### 5.2.10.38 TIMSTMP1MSB Register (Offset = 43Ch) [Reset = 00000000h]

TIMSTMP1MSB is shown in [Table 5-1271](#).

Return to the [Summary Table](#).

**Table 5-1271. TIMSTMP1MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP1MSB	R	0h	Time Stamp 1 MSB Count Output

### 5.2.10.39 TIMSTMP2LSB Register (Offset = 440h) [Reset = 00000000h]

TIMSTMP2LSB is shown in [Table 5-1272](#).

Return to the [Summary Table](#).

**Table 5-1272. TIMSTMP2LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP2LSB	R	0h	Time Stamp 2 LSB Count Output

#### 5.2.10.40 TIMSTMP2MID Register (Offset = 444h) [Reset = 00000000h]

TIMSTMP2MID is shown in [Table 5-1273](#).

Return to the [Summary Table](#).

**Table 5-1273. TIMSTMP2MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP2MID	R	0h	Time Stamp 2 MID Count Output

#### 5.2.10.41 TIMSTMP2MSB Register (Offset = 448h) [Reset = 00000000h]

TIMSTMP2MSB is shown in [Table 5-1274](#).

Return to the [Summary Table](#).

**Table 5-1274. TIMSTMP2MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP2MSB	R	0h	Time Stamp 2 MSB Count Output

#### 5.2.10.42 TIMSTMP3LSB Register (Offset = 44Ch) [Reset = 00000000h]

TIMSTMP3LSB is shown in [Table 5-1275](#).

Return to the [Summary Table](#).

**Table 5-1275. TIMSTMP3LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP3LSB	R	0h	Time Stamp 3 LSB Count Output

### 5.2.10.43 TIMSTMP3MID Register (Offset = 450h) [Reset = 00000000h]

TIMSTMP3MID is shown in [Table 5-1276](#).

Return to the [Summary Table](#).

**Table 5-1276. TIMSTMP3MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP3MID	R	0h	Time Stamp 3 MID Count Output



#### 5.2.10.44 TIMSTMP3MSB Register (Offset = 454h) [Reset = 00000000h]

TIMSTMP3MSB is shown in [Table 5-1277](#).

Return to the [Summary Table](#).

**Table 5-1277. TIMSTMP3MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP3MSB	R	0h	Time Stamp 3 MSB Count Output

### 5.2.10.45 TIMSTMP4LSB Register (Offset = 458h) [Reset = 00000000h]

TIMSTMP4LSB is shown in [Table 5-1278](#).

Return to the [Summary Table](#).

**Table 5-1278. TIMSTMP4LSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP4LSB	R	0h	Time Stamp 4 LSB Count Output

#### 5.2.10.46 TIMSTMP4MID Register (Offset = 45Ch) [Reset = 00000000h]

TIMSTMP4MID is shown in [Table 5-1279](#).

Return to the [Summary Table](#).

**Table 5-1279. TIMSTMP4MID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP4MID	R	0h	Time Stamp 4 MID Count Output

### 5.2.10.47 TIMSTMP4MSB Register (Offset = 460h) [Reset = 00000000h]

TIMSTMP4MSB is shown in [Table 5-1280](#).

Return to the [Summary Table](#).

**Table 5-1280. TIMSTMP4MSB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIMSTMP4MSB	R	0h	Time Stamp 4 MSB Count Output

#### 5.2.10.48 FRCLSBCNT Register (Offset = 464h) [Reset = 0FFFFFFFh]

FRCLSBCNT is shown in [Table 5-1281](#).

Return to the [Summary Table](#).

**Table 5-1281. FRCLSBCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FRCLSBCNT	R	0FFFFFFFh	LSB Count Value of FRC

### 5.2.10.49 FRCMIDCNT Register (Offset = 468h) [Reset = 00000000h]

FRCMIDCNT is shown in [Table 5-1282](#).

Return to the [Summary Table](#).

**Table 5-1282. FRCMIDCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FRCMIDCNT	R	0h	MID Count Value of FRC

### 5.2.10.50 FRCMSBCNT Register (Offset = 46Ch) [Reset = 00000000h]

FRCMSBCNT is shown in [Table 5-1283](#).

Return to the [Summary Table](#).

**Table 5-1283. FRCMSBCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FRCMSBCNT	R	0h	MSB Count Value of FRC

### 5.2.10.51 TIMSTMPTRIGSEL Register (Offset = 470h) [Reset = 00X0X0X0h]

TIMSTMPTRIGSEL is shown in [Table 5-1284](#).

Return to the [Summary Table](#).

**Table 5-1284. TIMSTMPTRIGSEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-24	TIMSTMPTRIG1SEL	R/W	0h	Trigger select for Time Stamp 1. 0-->MSS Sw Trig1, 1-->MSS SW Trig2, 2-->GPIO1, 3-->GPIO2, 4-->GPIO3, 5-->MSS MSG1, 6-->MSS MSG2, 7-->MSS MSG3, 8-->Ext Sync In, 9-->MSS MSG4, 10-->MSS MSG5
23-20	RESERVED	R/W	0h	
19-16	TIMSTMPTRIG2SEL	R/W	0h	Trigger select for Time Stamp 2. 0-->MSS Sw Trig1, 1-->MSS SW Trig2, 2-->GPIO1, 3-->GPIO2, 4-->GPIO3, 5-->MSS MSG1, 6-->MSS MSG2, 7-->MSS MSG3, 8-->Ext Sync In, 9-->MSS MSG4, 10-->MSS MSG5
15-12	RESERVED	R/W	0h	
11-8	TIMSTMPTRIG3SEL	R/W	0h	Trigger select for Time Stamp 3. 0-->MSS Sw Trig1, 1-->MSS SW Trig2, 2-->GPIO1, 3-->GPIO2, 4-->GPIO3, 5-->MSS MSG1, 6-->MSS MSG2, 7-->MSS MSG3, 8-->Ext Sync In, 9-->MSS MSG4, 10-->MSS MSG5
7-4	RESERVED	R/W	0h	
3-0	TIMSTMPTRIG4SEL	R/W	0h	Trigger select for Time Stamp 4. 0-->MSS Sw Trig1, 1-->MSS SW Trig2, 2-->GPIO1, 3-->GPIO2, 4-->GPIO3, 5-->MSS MSG1, 6-->MSS MSG2, 7-->MSS MSG3, 8-->Ext Sync In, 9-->MSS MSG4, 10-->MSS MSG5



### 5.2.10.52 FRCLSBPERIOD Register (Offset = 474h) [Reset = 00000000h]

FRCLSBPERIOD is shown in [Table 5-1285](#).

Return to the [Summary Table](#).

**Table 5-1285. FRCLSBPERIOD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	FRCLSBPERIOD	R/W	0h	LSB Period for FRC. This is copied into the shadow registers inside the module at the roll over condition and hence any update to these registers during the frame will take effect for the next frame.

### 5.2.10.53 FRCMIDPERIOD Register (Offset = 478h) [Reset = 00000000h]

FRCMIDPERIOD is shown in [Table 5-1286](#).

Return to the [Summary Table](#).

**Table 5-1286. FRCMIDPERIOD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	FRCMIDPERIOD	R/W	0h	MID Period for FRC. This is copied into the shadow registers inside the module at the roll over condition and hence any update to these registers during the frame will take effect for the next frame.

### 5.2.10.54 FRCLSBCNTINITVAL Register (Offset = 47Ch) [Reset = 0FFFFFFFh]

FRCLSBCNTINITVAL is shown in [Table 5-1287](#).

Return to the [Summary Table](#).

**Table 5-1287. FRCLSBCNTINITVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	FRCLSBCNTINITVAL	R/W	0FFFFFFFh	Init Value for FRC LSB Counter

### 5.2.10.55 FRCMIDCNTINITVAL Register (Offset = 480h) [Reset = 00000000h]

FRCMIDCNTINITVAL is shown in [Table 5-1288](#).

Return to the [Summary Table](#).

**Table 5-1288. FRCMIDCNTINITVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	FRCMIDCNTINITVAL	R/W	0h	Init Value for FRC MID Counter

### 5.2.10.56 FRCMSBCNTINITVAL Register (Offset = 484h) [Reset = 00000000h]

FRCMSBCNTINITVAL is shown in [Table 5-1289](#).

Return to the [Summary Table](#).

**Table 5-1289. FRCMSBCNTINITVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	FRCMSBCNTINITVAL	R/W	0h	Init Value for FRC MSB Counter

### 5.2.10.57 FRCSYNCOU TLSBCNT Register (Offset = 488h) [Reset = 00000000h]

FRCSYNCOU TLSBCNT is shown in [Table 5-1290](#).

Return to the [Summary Table](#).

**Table 5-1290. FRCSYNCOU TLSBCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	
27-0	FRCSYNCOU TLSBCNT	R/W	0h	LSB Count value used for generating Frame Pulse (ext_sync_out)

### 5.2.10.58 FRCSYNCOU TMIDCNT Register (Offset = 48Ch) [Reset = 00000000h]

FRCSYNCOU TMIDCNT is shown in [Table 5-1291](#).

Return to the [Summary Table](#).

**Table 5-1291. FRCSYNCOU TMIDCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	
11-0	FRCSYNCOU TMIDCNT	R/W	0h	MID Count value used for generating Frame Pulse (ext_sync_out)

### 5.2.10.59 FRCRESETMODE Register (Offset = 490h) [Reset = 00000000h]

FRCRESETMODE is shown in [Table 5-1292](#).

Return to the [Summary Table](#).

**Table 5-1292. FRCRESETMODE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	
1	FRCMIDRESETMODE	R/W	0h	Reset mode for MID Counter. 0-->Sync_In, 1-->LSB Period
0	FRCLSBRESETMODE	R/W	0h	Reset mode for LSB Counter. 0-->Sync_In, 1-->LSB Period



### 5.2.10.60 FRCCFG2 Register (Offset = 494h) [Reset = 00XX0010h]

FRCCFG2 is shown in [Table 5-1293](#).

Return to the [Summary Table](#).

**Table 5-1293. FRCCFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	
24	FRCTRIGMSS	R/W	0h	SW Trigger for FRC from MSS.Self clearing
23-18	RESERVED	R/W	0h	
17-16	FRCSYNCINSEL	R/W	0h	Mux select for External Frame Trigger source 0:Select SYNC_IN IO 1:Select RTID_INTRundefined 2:Select RTID_INTRundefined 3:Select MSS_FRCTRIGMSS
15-4	CANTRIGDLYVAL	R/W	1h	Delay value between the CAN interrupt and FRC trigger
3-0	CANINTRMASK	R/W	0h	Bit level mask for each of the CAN interrupts

### 5.2.10.61 HW\_SPARE\_RW0 Register (Offset = FD0h) [Reset = 00000000h]

HW\_SPARE\_RW0 is shown in [Table 5-1294](#).

Return to the [Summary Table](#).

**Table 5-1294. HW\_SPARE\_RW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw0	R/W	0h	Reserved for HW R&D

### 5.2.10.62 HW\_SPARE\_RW1 Register (Offset = FD4h) [Reset = 00000000h]

HW\_SPARE\_RW1 is shown in [Table 5-1295](#).

Return to the [Summary Table](#).

**Table 5-1295. HW\_SPARE\_RW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw1	R/W	0h	Reserved for HW R&D

### 5.2.10.63 HW\_SPARE\_RW2 Register (Offset = FD8h) [Reset = 00000000h]

HW\_SPARE\_RW2 is shown in [Table 5-1296](#).

Return to the [Summary Table](#).

**Table 5-1296. HW\_SPARE\_RW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw2	R/W	0h	Reserved for HW R&D

#### 5.2.10.64 HW\_SPARE\_RW3 Register (Offset = FDCh) [Reset = 00000000h]

HW\_SPARE\_RW3 is shown in [Table 5-1297](#).

Return to the [Summary Table](#).

**Table 5-1297. HW\_SPARE\_RW3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_rw3	R/W	0h	Reserved for HW R&D

### 5.2.10.65 HW\_SPARE\_RO0 Register (Offset = FE0h) [Reset = 00000000h]

HW\_SPARE\_RO0 is shown in [Table 5-1298](#).

Return to the [Summary Table](#).

**Table 5-1298. HW\_SPARE\_RO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro0	R	0h	Reserved for HW R&D

### 5.2.10.66 HW\_SPARE\_RO1 Register (Offset = FE4h) [Reset = 00000000h]

HW\_SPARE\_RO1 is shown in [Table 5-1299](#).

Return to the [Summary Table](#).

**Table 5-1299. HW\_SPARE\_RO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro1	R	0h	Reserved for HW R&D

### 5.2.10.67 HW\_SPARE\_RO2 Register (Offset = FE8h) [Reset = 00000000h]

HW\_SPARE\_RO2 is shown in [Table 5-1300](#).

Return to the [Summary Table](#).

**Table 5-1300. HW\_SPARE\_RO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro2	R	0h	Reserved for HW R&D



### 5.2.10.68 HW\_SPARE\_RO3 Register (Offset = FECh) [Reset = 00000000h]

HW\_SPARE\_RO3 is shown in [Table 5-1301](#).

Return to the [Summary Table](#).

**Table 5-1301. HW\_SPARE\_RO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_ro3	R	0h	Reserved for HW R&D

### 5.2.10.69 HW\_SPARE\_WPH Register (Offset = FF0h) [Reset = 00000000h]

HW\_SPARE\_WPH is shown in [Table 5-1302](#).

Return to the [Summary Table](#).

**Table 5-1302. HW\_SPARE\_WPH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	hw_spare_wph	R/W	0h	Reserved for HW R&D

### 5.2.10.70 HW\_SPARE\_REC Register (Offset = FF4h) [Reset = 00000000h]

HW\_SPARE\_REC is shown in [Table 5-1303](#).

Return to the [Summary Table](#).

**Table 5-1303. HW\_SPARE\_REC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	hw_spare_rec31	R/W	0h	Reserved for HW R&D
30	hw_spare_rec30	R/W	0h	Reserved for HW R&D
29	hw_spare_rec29	R/W	0h	Reserved for HW R&D
28	hw_spare_rec28	R/W	0h	Reserved for HW R&D
27	hw_spare_rec27	R/W	0h	Reserved for HW R&D
26	hw_spare_rec26	R/W	0h	Reserved for HW R&D
25	hw_spare_rec25	R/W	0h	Reserved for HW R&D
24	hw_spare_rec24	R/W	0h	Reserved for HW R&D
23	hw_spare_rec23	R/W	0h	Reserved for HW R&D
22	hw_spare_rec22	R/W	0h	Reserved for HW R&D
21	hw_spare_rec21	R/W	0h	Reserved for HW R&D
20	hw_spare_rec20	R/W	0h	Reserved for HW R&D
19	hw_spare_rec19	R/W	0h	Reserved for HW R&D
18	hw_spare_rec18	R/W	0h	Reserved for HW R&D
17	hw_spare_rec17	R/W	0h	Reserved for HW R&D
16	hw_spare_rec16	R/W	0h	Reserved for HW R&D
15	hw_spare_rec15	R/W	0h	Reserved for HW R&D
14	hw_spare_rec14	R/W	0h	Reserved for HW R&D
13	hw_spare_rec13	R/W	0h	Reserved for HW R&D
12	hw_spare_rec12	R/W	0h	Reserved for HW R&D
11	hw_spare_rec11	R/W	0h	Reserved for HW R&D
10	hw_spare_rec10	R/W	0h	Reserved for HW R&D
9	hw_spare_rec9	R/W	0h	Reserved for HW R&D
8	hw_spare_rec8	R/W	0h	Reserved for HW R&D
7	hw_spare_rec7	R/W	0h	Reserved for HW R&D
6	hw_spare_rec6	R/W	0h	Reserved for HW R&D
5	hw_spare_rec5	R/W	0h	Reserved for HW R&D
4	hw_spare_rec4	R/W	0h	Reserved for HW R&D
3	hw_spare_rec3	R/W	0h	Reserved for HW R&D
2	hw_spare_rec2	R/W	0h	Reserved for HW R&D
1	hw_spare_rec1	R/W	0h	Reserved for HW R&D
0	hw_spare_rec0	R/W	0h	Reserved for HW R&D

### 5.2.10.71 LOCK0\_KICK0 Register (Offset = 1008h) [Reset = 00000000h]

LOCK0\_KICK0 is shown in [Table 5-1304](#).

Return to the [Summary Table](#).

- KICK0 component

**Table 5-1304. LOCK0\_KICK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick0	R/W	0h	- KICK0 component

### 5.2.10.72 LOCK0\_KICK1 Register (Offset = 100Ch) [Reset = 00000000h]

LOCK0\_KICK1 is shown in [Table 5-1305](#).

Return to the [Summary Table](#).

- KICK1 component

**Table 5-1305. LOCK0\_KICK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LOCK0_kick1	R/W	0h	- KICK1 component

### 5.2.10.73 intr\_raw\_status Register (Offset = 1010h) [Reset = 00000000h]

intr\_raw\_status is shown in [Table 5-1306](#).

Return to the [Summary Table](#).

Interrupt Raw Status/Set Register

**Table 5-1306. intr\_raw\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err	R/W1S	0h	Proxy0 access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
2	kick_err	R/W1S	0h	Kick access violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
1	addr_err	R/W1S	0h	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	R/W1S	0h	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

### 5.2.10.74 intr\_enabled\_status\_clear Register (Offset = 1014h) [Reset = 00000000h]

intr\_enabled\_status\_clear is shown in [Table 5-1307](#).

Return to the [Summary Table](#).

Interrupt Enabled Status/Clear register

**Table 5-1307. intr\_enabled\_status\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	enabled_proxy_err	R/W1C	0h	Proxy0 access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
2	enabled_kick_err	R/W1C	0h	Kick access violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
1	enabled_addr_err	R/W1C	0h	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	R/W1C	0h	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

### 5.2.10.75 intr\_enable Register (Offset = 1018h) [Reset = 00000000h]

intr\_enable is shown in [Table 5-1308](#).

Return to the [Summary Table](#).

Interrupt Enable register

**Table 5-1308. intr\_enable Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en	R/W1S	0h	Proxy0 access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
2	kick_err_en	R/W1S	0h	Kick access violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
1	addr_err_en	R/W1S	0h	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	R/W1S	0h	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.



### 5.2.10.76 intr\_enable\_clear Register (Offset = 101Ch) [Reset = 00000000h]

intr\_enable\_clear is shown in [Table 5-1309](#).

Return to the [Summary Table](#).

Interrupt Enable Clear register

**Table 5-1309. intr\_enable\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	
3	proxy_err_en_clr	R/W1C	0h	Proxy0 access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
2	kick_err_en_clr	R/W1C	0h	Kick access violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
1	addr_err_en_clr	R/W1C	0h	Addressing violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	R/W1C	0h	Protection violation error enable clear. Write a 1 to clear the enable. Writing a 0 has no effect.

### 5.2.10.77 eoi Register (Offset = 1020h) [Reset = 00000000h]

eoi is shown in [Table 5-1310](#).

Return to the [Summary Table](#).

EOI register

**Table 5-1310. eoi Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	0h	
7-0	eoi_vector	R/W	0h	EOI vector value. Write this with interrupt distribution value in the chip.

### 5.2.10.78 fault\_address Register (Offset = 1024h) [Reset = 00000000h]

fault\_address is shown in [Table 5-1311](#).

Return to the [Summary Table](#).

Fault Address register

**Table 5-1311. fault\_address Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	fault_addr	R	0h	Fault Address.

### 5.2.10.79 fault\_type\_status Register (Offset = 1028h) [Reset = 00000000h]

fault\_type\_status is shown in [Table 5-1312](#).

Return to the [Summary Table](#).

Fault Type Status register

**Table 5-1312. fault\_type\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	fault_ns	R	0h	Non-secure access.
5-0	fault_type	R	0h	Fault Type 10_0000 = Supervisor read fault - priv = 1 dir = 1 dtype ! = 1 01_0000 = Supervisor write fault - priv = 1 dir = 0 00_1000 = Supervisor execute fault - priv = 1 dir = 1 dtype = 1 00_0100 = User read fault - priv = 0 dir = 1 dtype = 1 00_0010 = User write fault - priv = 0 dir = 0 00_0001 = User execute fault - priv = 0 dir = 1 dtype = 1 00_0000 = No fault

### 5.2.10.80 fault\_attr\_status Register (Offset = 102Ch) [Reset = 00000000h]

fault\_attr\_status is shown in [Table 5-1313](#).

Return to the [Summary Table](#).

Fault Attribute Status register

**Table 5-1313. fault\_attr\_status Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	fault_xid	R	0h	XID.
19-8	fault_routeid	R	0h	Route ID.
7-0	fault_privid	R	0h	Privilege ID.

### 5.2.10.81 fault\_clear Register (Offset = 1030h) [Reset = 00000000h]

fault\_clear is shown in [Table 5-1314](#).

Return to the [Summary Table](#).

Fault Clear register

**Table 5-1314. fault\_clear Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	W	0h	
0	fault_clr	W	0h	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

### 5.2.11 DSP\_ICFG Registers

Table 5-1315 lists the memory-mapped registers for the DSP\_ICFG registers. All register offset addresses not listed in Table 5-1315 should be considered as reserved locations and the register contents should not be modified.

**Table 5-1315. DSP\_ICFG Registers**

Offset	Acronym	Register Name	Section
0h	EVTFLAG0	Event flags	<a href="#">Go</a>
4h	EVTFLAG1	Event flags	<a href="#">Go</a>
8h	EVTFLAG2	Event flags	<a href="#">Go</a>
Ch	EVTFLAG3	Event flags	<a href="#">Go</a>
20h	EVTSET0	Event set command register	<a href="#">Go</a>
24h	EVTSET1	Event set command register	<a href="#">Go</a>
28h	EVTSET2	Event set command register	<a href="#">Go</a>
2Ch	EVTSET3	Event set command register	<a href="#">Go</a>
40h	EVTCLR0	Event clear command register	<a href="#">Go</a>
44h	EVTCLR1	Event clear command register	<a href="#">Go</a>
48h	EVTCLR2	Event clear command register	<a href="#">Go</a>
4Ch	EVTCLR3	Event clear command register	<a href="#">Go</a>
80h	EVTMASK0	Event mask register	<a href="#">Go</a>
84h	EVTMASK1	Event mask register	<a href="#">Go</a>
88h	EVTMASK2	Event mask register	<a href="#">Go</a>
8Ch	EVTMASK3	Event mask register	<a href="#">Go</a>
A0h	MEVTFLAG0	Masked event flags	<a href="#">Go</a>
A4h	MEVTFLAG1	Masked event flags	<a href="#">Go</a>
A8h	MEVTFLAG2	Masked event flags	<a href="#">Go</a>
ACh	MEVTFLAG3	Masked event flags	<a href="#">Go</a>
C0h	EXPMASK0	Exception mask register	<a href="#">Go</a>
C4h	EXPMASK1	Exception mask register	<a href="#">Go</a>
C8h	EXPMASK2	Exception mask register	<a href="#">Go</a>
CCh	EXPMASK3	Exception mask register	<a href="#">Go</a>
E0h	MEXPFLAG0	Masked exception flags	<a href="#">Go</a>
E4h	MEXPFLAG1	Masked exception flags	<a href="#">Go</a>
E8h	MEXPFLAG2	Masked exception flags	<a href="#">Go</a>
ECh	MEXPFLAG3	Masked exception flags	<a href="#">Go</a>
104h	INTMUX1	Interrupt Mux Register	<a href="#">Go</a>
108h	INTMUX2	Interrupt Mux Register	<a href="#">Go</a>
10Ch	INTMUX3	Interrupt Mux Register	<a href="#">Go</a>
140h	AEGMUX0	AEG Event Mux Register	<a href="#">Go</a>
144h	AEGMUX1	AEG Event Mux Register	<a href="#">Go</a>
180h	INTXSTAT	Interrupt Exception Status Register	<a href="#">Go</a>
184h	INTXCLR	Interrupt Exception Clear Register	<a href="#">Go</a>
188h	INTDMASK	Dropped Interrupt Mask Register	<a href="#">Go</a>
1C0h	EVTASRT	Event Assert Register	<a href="#">Go</a>
00010000h	PDCCMD		<a href="#">Go</a>
00012000h	MM_REVID		<a href="#">Go</a>
00020000h	IDMA0_STAT	IDMA Channel 0 Status	<a href="#">Go</a>

**Table 5-1315. DSP\_ICFG Registers (continued)**

Offset	Acronym	Register Name	Section
00020004h	IDMA0_MASK	IDMA Channel 0 Register Mask	<a href="#">Go</a>
00020008h	IDMA0_SOURCE	IDMA Channel 0 Source Address	<a href="#">Go</a>
0002000Ch	IDMA0_DEST	IDMA Channel 0 Destination Address	<a href="#">Go</a>
00020010h	IDMA0_COUNT	IDMA Channel 0 Count	<a href="#">Go</a>
00020100h	IDMA1_STAT		<a href="#">Go</a>
00020108h	IDMA1_SOURCE		<a href="#">Go</a>
0002010Ch	IDMA1_DEST		<a href="#">Go</a>
00020110h	IDMA1_COUNT		<a href="#">Go</a>
00020200h	CPUARBE		<a href="#">Go</a>
00020204h	IDMAARBE		<a href="#">Go</a>
00020208h	SDMAARBE		<a href="#">Go</a>
00020210h	ECFGARBE		<a href="#">Go</a>
00020300h	ICFGMPFAR		<a href="#">Go</a>
00020304h	ICFGMPFSR		<a href="#">Go</a>
00020308h	ICFGMPFCR		<a href="#">Go</a>
00020408h	ECFGERR		<a href="#">Go</a>
0002040Ch	ECFGERRCLR		<a href="#">Go</a>
00020500h	PAMAP0	PAMAP register	<a href="#">Go</a>
00020504h	PAMAP1	PAMAP register	<a href="#">Go</a>
00020508h	PAMAP2	PAMAP register	<a href="#">Go</a>
0002050Ch	PAMAP3	PAMAP register	<a href="#">Go</a>
00020510h	PAMAP4	PAMAP register	<a href="#">Go</a>
00020514h	PAMAP5	PAMAP register	<a href="#">Go</a>
00020518h	PAMAP6	PAMAP register	<a href="#">Go</a>
0002051Ch	PAMAP7	PAMAP register	<a href="#">Go</a>
00020520h	PAMAP8	PAMAP register	<a href="#">Go</a>
00020524h	PAMAP9	PAMAP register	<a href="#">Go</a>
00020528h	PAMAP10	PAMAP register	<a href="#">Go</a>
0002052Ch	PAMAP11	PAMAP register	<a href="#">Go</a>
00020530h	PAMAP12	PAMAP register	<a href="#">Go</a>
00020534h	PAMAP13	PAMAP register	<a href="#">Go</a>
00020538h	PAMAP14	PAMAP register	<a href="#">Go</a>
0002053Ch	PAMAP15	PAMAP register	<a href="#">Go</a>
00031100h	EDCINTMASK	This read write registers contains the masks for the exceptions generated by each of the EDC error detection units added in this change. Also contained in this register are the bit fields indicating which module has issued an exception. Reads of this register do not affect the mask bits but do clear the exception bits.	<a href="#">Go</a>
00031104h	EDCINTFLG	This register loads the interrupt flags upon issue by the EDC detection module. These flags are loaded independently of any other flags. Upon a read operation all flags are cleared.	<a href="#">Go</a>



**Table 5-1315. DSP\_ICFG Registers (continued)**

Offset	Acronym	Register Name	Section
00031108h	L1DEDCMD	<p>This write only register provides control for enabling and disabling the ECC logic of the DMC.</p> <p>This register controls the parameters which are accumulated for the Data and Tag RAMs in the DMC. Enabling the ECC allows the counting of recoverable and non-recoverable errors as well as registering the requestor and the address and cache states for any ECC errors which are reported.</p>	<a href="#">Go</a>
0003110Ch	L1DDCSTAT	<p>This read only register contains the current error data for Correctable ECC errors.</p> <p>The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.</p> <p>The error information is written upon the occurrence of the first ECC error encountered. CERR1 is set, and the values are retained. If any additional ECC errors occur, this is indicated by setting CERR2.</p>	<a href="#">Go</a>
00031110h	L1DDNCSTAT	<p>This read only register contains the current error data for Correctable ECC errors.</p> <p>The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.</p> <p>This register is similar to Error! Not a valid bookmark self-reference. with the exception of the bit position register fields, with two bit errors it is not possible to provide the locations.</p> <p>The error information is written upon the occurrence of the first ECC error encountered. CERR1 is set, and the values are retained. If any additional ECC errors occur, this is indicated by setting CERR2.</p> <p>Note: Non-correctable errors do not mean two bits in the 256 bit data word, but two bits+ of error in a single byte.</p> <p>The EDC is capable of correcting up to 8 bits in a single word if there are only single bit errors in each byte.</p>	<a href="#">Go</a>
00031114h	L1DTCSTAT	<p>This read only register contains the current error data for Correctable ECC errors.</p> <p>The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.</p> <p>The error information is written upon the occurrence of the first ECC error encountered. CERR1 is set, and the values are retained. If any additional ECC errors occur, this is indicated by setting CERR2.</p>	<a href="#">Go</a>
00031118h	L1DTNCSTAT	L1D Error Detect TAG Non-Correctable Status Register	<a href="#">Go</a>

**Table 5-1315. DSP\_ICFG Registers (continued)**

Offset	Acronym	Register Name	Section
0003111Ch	L1DDEDADDR	The L1D Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in the STATUS registers. The addresses for both correctable and noncorrectable errors are held in this register. The value contained in the address field is the lower 16 bits of the word boundary of the memory access. If the access is to 0x00F00020, the address field will read 0x0020, if the access is to 0x00F07FE0, the address field will read 0x7FE0.	<a href="#">Go</a>
00031120h	L1DTEDADDR	The L1D Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in the STATUS registers. The address for both correctable and noncorrectable errors are held in this register. The value of the address is the index (or address on the RAM) of the RAM module.	<a href="#">Go</a>
00031124h	L1DEDCNT	The L1D TAG Error Detection Error Count Register stores the error counts for correctable and noncorrectable errors which occur in the L1D Cache Memories, these counts are accumulated regardless of the exception mapping in the L1DEDCMD (0x0182_1108) interrupt enable bits. The counters capture the total number of correctable and non-correctable errors from both the DATA and TAG memories.	<a href="#">Go</a>
00031128h	L2TEDCMD	This write only register provides control for enabling and disabling the ECC logic off the UMC. This register controls the parameters which are accumulated for the Data and Tag RAMs in the DMC. Enabling the ECC allows the counting of recoverable and non-recoverable errors as well as registering the requestor and the address and cache states for any ECC errors which are reported.	<a href="#">Go</a>
0003112Ch	L2TCSTAT	This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>
00031130h	L2TNCSTAT	This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>
00031134h	L2TEDADDR	The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions. The address in these fields is the displacement into the memory as presented to the address port of the SRAM.	<a href="#">Go</a>
0003113Ch	L2MNCSTAT	This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>

**Table 5-1315. DSP\_ICFG Registers (continued)**

Offset	Acronym	Register Name	Section
00031140h	L2MEDADDR	The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions.	<a href="#">Go</a>
00031144h	L2SCSTAT	This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>
00031148h	L2SNCSTAT	This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>
0003114Ch	L2SEDADDR	The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions.	<a href="#">Go</a>
00031150h	L2LCSTAT	This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>
00031154h	L2LNCSTAT		<a href="#">Go</a>
00031158h	L2LEDADDR	The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions.	<a href="#">Go</a>
0003115Ch	L2TEDCNT	The L2 Error Detection Error Count Register stores the error counts for correctable and non-correctable errors which occur in the L1D Data Memory, these counts are accumulated regardless of the exception mapping in the L1DEDCMD (0x0182_1108) interrupt enable bits. The counts contained in these registers are the accumulation of all L2 errors. This value saturates at an unsigned 0xFFFF.	<a href="#">Go</a>
00031160h	L1PTEDCMD	This write only register provides control for enabling and disabling the Parity logic of the PMC. This register controls the parameters which are accumulated for the Data and Tag RAMs in the PMC. Enabling the Parity allows the counting of recoverable and non-recoverable errors as well as registering the requestor and the address and cache states for any Parity errors which are reported.	<a href="#">Go</a>
00031164h	L1PTEDSTAT	This read only register contains the current error. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.	<a href="#">Go</a>

**Table 5-1315. DSP\_ICFG Registers (continued)**

Offset	Acronym	Register Name	Section
00031168h	L1PTEDADDR	The address error registers for correctable and non-correctable errors are identical in structure, the indication applies to the correctable and non-correctable errors, they are both covered in this section. The register provides a record of the address of the detected error and the access status or the cache.	<a href="#">Go</a>
0003116Ch	L1DTEDCNT	This read only register contains the counts of correctable and non-correctable errors which have occurred. To reset these counts write to the DMCEDCMD register in the appropriate bit.	<a href="#">Go</a>
00040000h	L2CFG		<a href="#">Go</a>
00040020h	L1PCFG		<a href="#">Go</a>
00040024h	L1PCC		<a href="#">Go</a>
00040040h	L1DCFG		<a href="#">Go</a>
00040044h	L1DCC		<a href="#">Go</a>
00041000h	CPUARBU		<a href="#">Go</a>
00041004h	IDMAARBU		<a href="#">Go</a>
00041008h	SDMAARBU		<a href="#">Go</a>
0004100Ch	UCARBU		<a href="#">Go</a>
00041010h	MDMAARBU		<a href="#">Go</a>
00041040h	CPUARBD		<a href="#">Go</a>
00041044h	IDMAARBD		<a href="#">Go</a>
00041048h	SDMAARBD		<a href="#">Go</a>
0004104Ch	UCARBD		<a href="#">Go</a>
00044000h	L2WBAR		<a href="#">Go</a>
00044004h	L2WWC		<a href="#">Go</a>
00044010h	L2WIBAR		<a href="#">Go</a>
00044014h	L2WIWC		<a href="#">Go</a>
00044018h	L2IBAR		<a href="#">Go</a>
0004401Ch	L2IWC		<a href="#">Go</a>
00044020h	L1PIBAR		<a href="#">Go</a>
00044024h	L1PIWC		<a href="#">Go</a>
00044030h	L1DWIBAR		<a href="#">Go</a>
00044034h	L1DWIWC		<a href="#">Go</a>
00044040h	L1DWBAR		<a href="#">Go</a>
00044044h	L1DWWC		<a href="#">Go</a>
00044048h	L1DIBAR		<a href="#">Go</a>
0004404Ch	L1DIWC		<a href="#">Go</a>
00045000h	L2WB		<a href="#">Go</a>
00045004h	L2WBINV		<a href="#">Go</a>
00045008h	L2INV		<a href="#">Go</a>
00045028h	L1PINV		<a href="#">Go</a>
00045040h	L1DWB		<a href="#">Go</a>
00045044h	L1DWBINV		<a href="#">Go</a>
00045048h	L1DINV		<a href="#">Go</a>
00046004h	L2EDSTAT	L2 EDC Status register	<a href="#">Go</a>
00046008h	L2EDCMD	L2 EDC Command register	<a href="#">Go</a>

**Table 5-1315. DSP\_ICFG Registers (continued)**

Offset	Acronym	Register Name	Section
0004600Ch	L2EDADDR	L2 EDC error address	<a href="#">Go</a>
00046018h	L2EDCPEC	L2 EDC correctable error count	<a href="#">Go</a>
0004601Ch	L2EDCNEC	L2 EDC noncorrectable error count	<a href="#">Go</a>
00046020h	MDMAERR		<a href="#">Go</a>
00046024h	MDMAERRCLR		<a href="#">Go</a>
00046030h	L2EDCEN	L2 EDC enable register	<a href="#">Go</a>
00046404h	L1PEDSTAT	L1P EDC Status register	<a href="#">Go</a>
00046408h	L1PEDCMD	L1P EDC Command register	<a href="#">Go</a>
0004640Ch	L1PEDADDR	L1P EDC error address	<a href="#">Go</a>
00048000h	MAR	MAR register	<a href="#">Go</a>
0004A000h	L2MPFAR		<a href="#">Go</a>
0004A004h	L2MPFSR		<a href="#">Go</a>
0004A008h	L2MPFCR		<a href="#">Go</a>
0004A200h	L2MPPA	L2 Memory Protection Page Attribute register	<a href="#">Go</a>
0004A400h	L1PMPFAR		<a href="#">Go</a>
0004A404h	L1PMPFSR		<a href="#">Go</a>
0004A408h	L1PMPFCR		<a href="#">Go</a>
0004A640h	L1PMPPA	L1P Memory Protection Page Attribute register	<a href="#">Go</a>
0004AC00h	L1DMPFAR		<a href="#">Go</a>
0004AC04h	L1DMPFSR		<a href="#">Go</a>
0004AC08h	L1DMPFCR		<a href="#">Go</a>
0004AD00h	MPLK	MPLK register	<a href="#">Go</a>
0004AD10h	MPLKCMD		<a href="#">Go</a>
0004AD14h	MPLKSTAT		<a href="#">Go</a>
0004AE40h	L1DMPPA	L1D Memory Protection Page Attribute register	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 5-1316](#) shows the codes that are used for access types in this section.

**Table 5-1316. DSP\_ICFG Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 5.2.11.1 EVTFLAG0 Register (Offset = 0h) [Reset = 00000000h]

EVTFLAG0 is shown in [Table 5-1317](#).

Return to the [Summary Table](#).

Event flags

**Table 5-1317. EVTFLAG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EF	R	0h	

### 5.2.11.2 EVTFLAG1 Register (Offset = 4h) [Reset = 00000000h]

EVTFLAG1 is shown in [Table 5-1318](#).

Return to the [Summary Table](#).

Event flags

**Table 5-1318. EVTFLAG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EF	R	0h	

### 5.2.11.3 EVTFLAG2 Register (Offset = 8h) [Reset = 00000000h]

EVTFLAG2 is shown in [Table 5-1319](#).

Return to the [Summary Table](#).

Event flags

**Table 5-1319. EVTFLAG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EF	R	0h	



#### 5.2.11.4 EVTFLAG3 Register (Offset = Ch) [Reset = 00000000h]

EVTFLAG3 is shown in [Table 5-1320](#).

Return to the [Summary Table](#).

Event flags

**Table 5-1320. EVTFLAG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EF	R	0h	

### 5.2.11.5 EVTSET0 Register (Offset = 20h) [Reset = 00000000h]

EVTSET0 is shown in [Table 5-1321](#).

Return to the [Summary Table](#).

Event set command register

**Table 5-1321. EVTSET0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ES	W	0h	Event set command

### 5.2.11.6 EVTSET1 Register (Offset = 24h) [Reset = 00000000h]

EVTSET1 is shown in [Table 5-1322](#).

Return to the [Summary Table](#).

Event set command register

**Table 5-1322. EVTSET1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ES	W	0h	Event set command

### 5.2.11.7 EVTSET2 Register (Offset = 28h) [Reset = 00000000h]

EVTSET2 is shown in [Table 5-1323](#).

Return to the [Summary Table](#).

Event set command register

**Table 5-1323. EVTSET2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ES	W	0h	Event set command

### 5.2.11.8 EVTSET3 Register (Offset = 2Ch) [Reset = 00000000h]

EVTSET3 is shown in [Table 5-1324](#).

Return to the [Summary Table](#).

Event set command register

**Table 5-1324. EVTSET3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ES	W	0h	Event set command

### 5.2.11.9 EVTCLR0 Register (Offset = 40h) [Reset = 00000000h]

EVTCLR0 is shown in [Table 5-1325](#).

Return to the [Summary Table](#).

Event clear command register

**Table 5-1325. EVTCLR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EC	W	0h	Event clear command

### 5.2.11.10 EVTCLR1 Register (Offset = 44h) [Reset = 00000000h]

EVTCLR1 is shown in [Table 5-1326](#).

Return to the [Summary Table](#).

Event clear command register

**Table 5-1326. EVTCLR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EC	W	0h	Event clear command

### 5.2.11.11 EVTCLR2 Register (Offset = 48h) [Reset = 00000000h]

EVTCLR2 is shown in [Table 5-1327](#).

Return to the [Summary Table](#).

Event clear command register

**Table 5-1327. EVTCLR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EC	W	0h	Event clear command



### 5.2.11.12 EVTCLR3 Register (Offset = 4Ch) [Reset = 00000000h]

EVTCLR3 is shown in [Table 5-1328](#).

Return to the [Summary Table](#).

Event clear command register

**Table 5-1328. EVTCLR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EC	W	0h	Event clear command

### 5.2.11.13 EVTMASK0 Register (Offset = 80h) [Reset = 00000000h]

EVTMASK0 is shown in [Table 5-1329](#).

Return to the [Summary Table](#).

Event mask register

**Table 5-1329. EVTMASK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EM	R/W	0h	Event mask

#### 5.2.11.14 EVTMASK1 Register (Offset = 84h) [Reset = 00000000h]

EVTMASK1 is shown in [Table 5-1330](#).

Return to the [Summary Table](#).

Event mask register

**Table 5-1330. EVTMASK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EM	R/W	0h	Event mask

### 5.2.11.15 EVTMASK2 Register (Offset = 88h) [Reset = 00000000h]

EVTMASK2 is shown in [Table 5-1331](#).

Return to the [Summary Table](#).

Event mask register

**Table 5-1331. EVTMASK2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EM	R/W	0h	Event mask

### 5.2.11.16 EVTMASK3 Register (Offset = 8Ch) [Reset = 00000000h]

EVTMASK3 is shown in [Table 5-1332](#).

Return to the [Summary Table](#).

Event mask register

**Table 5-1332. EVTMASK3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	EM	R/W	0h	Event mask

### 5.2.11.17 MEVTFLAG0 Register (Offset = A0h) [Reset = 00000000h]

MEVTFLAG0 is shown in [Table 5-1333](#).

Return to the [Summary Table](#).

Masked event flags

**Table 5-1333. MEVTFLAG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEVTFLAG	R	0h	Masked event flags

### 5.2.11.18 MEVTFLAG1 Register (Offset = A4h) [Reset = 00000000h]

MEVTFLAG1 is shown in [Table 5-1334](#).

Return to the [Summary Table](#).

Masked event flags

**Table 5-1334. MEVTFLAG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEVTFLAG	R	0h	Masked event flags

### 5.2.11.19 MEVTFLAG2 Register (Offset = A8h) [Reset = 00000000h]

MEVTFLAG2 is shown in [Table 5-1335](#).

Return to the [Summary Table](#).

Masked event flags

**Table 5-1335. MEVTFLAG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEVTFLAG	R	0h	Masked event flags



### 5.2.11.20 MEVTFLAG3 Register (Offset = ACh) [Reset = 00000000h]

MEVTFLAG3 is shown in [Table 5-1336](#).

Return to the [Summary Table](#).

Masked event flags

**Table 5-1336. MEVTFLAG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEVTFLAG	R	0h	Masked event flags

### 5.2.11.21 EXPMASK0 Register (Offset = C0h) [Reset = FFFFFFFFh]

EXPMASK0 is shown in [Table 5-1337](#).

Return to the [Summary Table](#).

Exception mask register

**Table 5-1337. EXPMASK0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	XM	R/W	FFFFFFFh	Exception mask

### 5.2.11.22 EXPMASK1 Register (Offset = C4h) [Reset = FFFFFFFFh]

EXPMASK1 is shown in [Table 5-1338](#).

Return to the [Summary Table](#).

Exception mask register

**Table 5-1338. EXPMASK1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	XM	R/W	FFFFFFFh	Exception mask

### 5.2.11.23 EXPMASK2 Register (Offset = C8h) [Reset = FFFFFFFFh]

EXPMASK2 is shown in [Table 5-1339](#).

Return to the [Summary Table](#).

Exception mask register

**Table 5-1339. EXPMASK2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	XM	R/W	FFFFFFFh	Exception mask

#### 5.2.11.24 EXPMASK3 Register (Offset = CCh) [Reset = FFFFFFFFh]

EXPMASK3 is shown in [Table 5-1340](#).

Return to the [Summary Table](#).

Exception mask register

**Table 5-1340. EXPMASK3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	XM	R/W	FFFFFFFh	Exception mask

### 5.2.11.25 MEXPFLAG0 Register (Offset = E0h) [Reset = 00000000h]

MEXPFLAG0 is shown in [Table 5-1341](#).

Return to the [Summary Table](#).

Masked exception flags

**Table 5-1341. MEXPFLAG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEXPFLAG	R	0h	Masked exception flags

### 5.2.11.26 MEXPFLAG1 Register (Offset = E4h) [Reset = 00000000h]

MEXPFLAG1 is shown in [Table 5-1342](#).

Return to the [Summary Table](#).

Masked exception flags

**Table 5-1342. MEXPFLAG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEXPFLAG	R	0h	Masked exception flags

### 5.2.11.27 MEXPFLAG2 Register (Offset = E8h) [Reset = 00000000h]

MEXPFLAG2 is shown in [Table 5-1343](#).

Return to the [Summary Table](#).

Masked exception flags

**Table 5-1343. MEXPFLAG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEXPFLAG	R	0h	Masked exception flags



### 5.2.11.28 MEXPFLAG3 Register (Offset = ECh) [Reset = 00000000h]

MEXPFLAG3 is shown in [Table 5-1344](#).

Return to the [Summary Table](#).

Masked exception flags

**Table 5-1344. MEXPFLAG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MEXPFLAG	R	0h	Masked exception flags

### 5.2.11.29 INTMUX1 Register (Offset = 104h) [Reset = 07060504h]

INTMUX1 is shown in [Table 5-1345](#).

Return to the [Summary Table](#).

Interrupt Mux Register

**Table 5-1345. INTMUX1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	_RESV3	0h	Always read as 0. Writes have no affect.	Interrupt selector 7
30-24	INTSEL7	R/W	7h	
23	_RESV2	0h	Always read as 0. Writes have no affect.	Interrupt selector 6
22-16	INTSEL6	R/W	6h	
15	_RESV1	0h	Always read as 0. Writes have no affect.	Interrupt selector 5
14-8	INTSEL5	R/W	5h	
7	_RESV0	0h	Always read as 0. Writes have no affect.	Interrupt selector 4
6-0	INTSEL4	R/W	4h	

### 5.2.11.30 INTMUX2 Register (Offset = 108h) [Reset = 0B0A0908h]

INTMUX2 is shown in [Table 5-1346](#).

Return to the [Summary Table](#).

Interrupt Mux Register

**Table 5-1346. INTMUX2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	_RESV3	0h	Always read as 0. Writes have no affect.	Interrupt selector 11
30-24	INTSEL11	R/W	Bh	
23	_RESV2	0h	Always read as 0. Writes have no affect.	Interrupt selector 10
22-16	INTSEL10	R/W	Ah	
15	_RESV1	0h	Always read as 0. Writes have no affect.	Interrupt selector 9
14-8	INTSEL9	R/W	9h	
7	_RESV0	0h	Always read as 0. Writes have no affect.	Interrupt selector 8
6-0	INTSEL8	R/W	8h	

### 5.2.11.31 INTMUX3 Register (Offset = 10Ch) [Reset = 0F0E0D0Ch]

INTMUX3 is shown in [Table 5-1347](#).

Return to the [Summary Table](#).

Interrupt Mux Register

**Table 5-1347. INTMUX3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	_RESV3	0h	Always read as 0. Writes have no affect.	Interrupt selector 15
30-24	INTSEL15	R/W	Fh	
23	_RESV2	0h	Always read as 0. Writes have no affect.	Interrupt selector 14
22-16	INTSEL14	R/W	Eh	
15	_RESV1	0h	Always read as 0. Writes have no affect.	Interrupt selector 13
14-8	INTSEL13	R/W	Dh	
7	_RESV0	0h	Always read as 0. Writes have no affect.	Interrupt selector 12
6-0	INTSEL12	R/W	Ch	

### 5.2.11.32 AEGMUX0 Register (Offset = 140h) [Reset = 03020100h]

AEGMUX0 is shown in [Table 5-1348](#).

Return to the [Summary Table](#).

AEG Event Mux Register

**Table 5-1348. AEGMUX0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-24	AEGSEL3	R/W	3h	AEG event selector 3
23	RESERVED	R	0h	
22-16	AEGSEL2	R/W	2h	AEG event selector 2
15	RESERVED	R	0h	
14-8	AEGSEL1	R/W	1h	AEG event selector 1
7	RESERVED	R	0h	
6-0	AEGSEL0	R/W	0h	AEG event selector 0

### 5.2.11.33 AEGMUX1 Register (Offset = 144h) [Reset = 07060504h]

AEGMUX1 is shown in [Table 5-1349](#).

Return to the [Summary Table](#).

AEG Event Mux Register

**Table 5-1349. AEGMUX1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-24	AEGSEL7	R/W	7h	AEG event selector 7
23	RESERVED	R	0h	
22-16	AEGSEL6	R/W	6h	AEG event selector 6
15	RESERVED	R	0h	
14-8	AEGSEL5	R/W	5h	AEG event selector 5
7	RESERVED	R	0h	
6-0	AEGSEL4	R/W	4h	AEG event selector 4

### 5.2.11.34 INTXSTAT Register (Offset = 180h) [Reset = 00000000h]

INTXSTAT is shown in [Table 5-1350](#).

Return to the [Summary Table](#).

Interrupt Exception Status Register

**Table 5-1350. INTXSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	SYSINT	R	0h	System event number
23-16	CPUINT	R	0h	CPU interrupt number
15-1	_RESV3	0h	Always read as 0. Writes have no affect.	
0	DROP	R	0h	Dropped event flag

### 5.2.11.35 INTXCLR Register (Offset = 184h) [Reset = 00000000h]

INTXCLR is shown in [Table 5-1351](#).

Return to the [Summary Table](#).

Interrupt Exception Clear Register

**Table 5-1351. INTXCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	_RESV3	0h	Always read as 0. Writes have no affect.	
0	CLEAR	W	0h	Clear interrupt exception status



### 5.2.11.36 INTDMASK Register (Offset = 188h) [Reset = 00000000h]

INTDMASK is shown in [Table 5-1352](#).

Return to the [Summary Table](#).

Dropped Interrupt Mask Register

**Table 5-1352. INTDMASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	_RESV1	0h	Always read as 0. Writes have no affect.	Interrupt drop mask
15-4	IDM	R/W	0h	
3-0	_RESV0	0h	Always read as 0. Writes have no affect.	

### 5.2.11.37 EVTASRT Register (Offset = 1C0h) [Reset = 03020100h]

EVTASRT is shown in [Table 5-1353](#).

Return to the [Summary Table](#).

Event Assert Register

**Table 5-1353. EVTASRT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	MRA	W	0h	Maximal Reset Request Assert
30	SRA	W	0h	System Reset Request Assert
29	CRA	W	0h	Chip Reset Request Assert
28	LRA	W	0h	Local Reset Request Assert
27-8	_RESV0	0h	Always read as 0. Writes have no affect.	
7-0	EA	W	0h	Event Assert

### 5.2.11.38 PDCCMD Register (Offset = 00010000h) [Reset = 00000000h]

PDCCMD is shown in [Table 5-1354](#).

Return to the [Summary Table](#).

**Table 5-1354. PDCCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESV_1	0h	Always read as 0. Writes have no affect.	
16	GEMPD	R/W	0h	Enable power-down during IDLE
15-14	RESV_2	0h	Always read as 0. Writes have no affect.	
13-12	EMCLOG	R/W	0h	EMC SRAM Sleep Mode
11-10	RESV_3	0h	Always read as 0. Writes have no affect.	
9-8	UMCLOG	R/W	0h	UMC SRAM Sleep Mode
7-6	DMCMEM	R/W	0h	DMC Logic Clock Gating
5-4	DMCLOG	R/W	0h	DMC SRAM Sleep Mode
3-2	PMCMEM	R/W	0h	PMC Logic Clock Gating
1-0	PMCLOG	R/W	0h	PMC SRAM Sleep Mode

### 5.2.11.39 MM\_REVID Register (Offset = 00012000h) [Reset = 00000000h]

MM\_REVID is shown in [Table 5-1355](#).

Return to the [Summary Table](#).

**Table 5-1355. MM\_REVID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MM_REVID	R/W	0h	

### 5.2.11.40 IDMA0\_STAT Register (Offset = 00020000h) [Reset = 00000000h]

IDMA0\_STAT is shown in [Table 5-1356](#).

Return to the [Summary Table](#).

IDMA Channel 0 Status

**Table 5-1356. IDMA0\_STAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	_RESV1	0h	Always read as 0. Writes have no affect.	
1	PEND	R	0h	Command pending
0	ACTV	R	0h	Command active

#### 5.2.11.41 IDMA0\_MASK Register (Offset = 00020004h) [Reset = 00000000h]

IDMA0\_MASK is shown in [Table 5-1357](#).

Return to the [Summary Table](#).

IDMA Channel 0 Register Mask

**Table 5-1357. IDMA0\_MASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	Register mask

#### 5.2.11.42 IDMA0\_SOURCE Register (Offset = 00020008h) [Reset = 00000000h]

IDMA0\_SOURCE is shown in [Table 5-1358](#).

Return to the [Summary Table](#).

IDMA Channel 0 Source Address

**Table 5-1358. IDMA0\_SOURCE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	SOURCEADDR	R/W	0h	Source address
4-0	_RESV1	0h	Always read as 0. Writes have no affect.	

### 5.2.11.43 IDMA0\_DEST Register (Offset = 0002000Ch) [Reset = 00000000h]

IDMA0\_DEST is shown in [Table 5-1359](#).

Return to the [Summary Table](#).

IDMA Channel 0 Destination Address

**Table 5-1359. IDMA0\_DEST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	DESTADDR	R/W	0h	Destination address
4-0	_RESV1	0h	Always read as 0. Writes have no affect.	



#### 5.2.11.44 IDMA0\_COUNT Register (Offset = 00020010h) [Reset = 00000000h]

IDMA0\_COUNT is shown in [Table 5-1360](#).

Return to the [Summary Table](#).

IDMA Channel 0 Count

**Table 5-1360. IDMA0\_COUNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	_RESV1	0h	Always read as 0. Writes have no affect.	Interrupt enable
28	INT	R/W	0h	
27-4	_RESV2	0h	Always read as 0. Writes have no affect.	Transfer frame count
3-0	COUNT	R/W	0h	

### 5.2.11.45 IDMA1\_STAT Register (Offset = 00020100h) [Reset = 00000000h]

IDMA1\_STAT is shown in [Table 5-1361](#).

Return to the [Summary Table](#).

**Table 5-1361. IDMA1\_STAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	_RESV1	0h	Always read as 0. Writes have no affect.	
1	PEND	R	0h	Command pending
0	ACTV	R	0h	Command active

### 5.2.11.46 IDMA1\_SOURCE Register (Offset = 00020108h) [Reset = 00000000h]

IDMA1\_SOURCE is shown in [Table 5-1362](#).

Return to the [Summary Table](#).

**Table 5-1362. IDMA1\_SOURCE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	SOURCEADDR	R/W	0h	Source address
1-0	_RESV1	0h	Always read as 0. Writes have no affect.	

### 5.2.11.47 IDMA1\_DEST Register (Offset = 0002010Ch) [Reset = 00000000h]

IDMA1\_DEST is shown in [Table 5-1363](#).

Return to the [Summary Table](#).

**Table 5-1363. IDMA1\_DEST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	DESTADDR	R/W	0h	Destination address
1-0	_RESV1	0h	Always read as 0. Writes have no affect.	

### 5.2.11.48 IDMA1\_COUNT Register (Offset = 00020110h) [Reset = 00000000h]

IDMA1\_COUNT is shown in [Table 5-1364](#).

Return to the [Summary Table](#).

**Table 5-1364. IDMA1\_COUNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	PRI	R/W	0h	Priority
28	INT	R/W	0h	Interrupt enable
27-17	_RESV1	0h	Always read as 0. Writes have no affect.	
16	FILL	R/W	0h	Fill/Copy Mode Select
15-2	COUNT	R/W	0h	Transfer word count
1-0	_RESV2	0h	Always read as 0. Writes have no affect.	

### 5.2.11.49 CPUARBE Register (Offset = 00020200h) [Reset = 00010010h]

CPUARBE is shown in [Table 5-1365](#).

Return to the [Summary Table](#).

**Table 5-1365. CPUARBE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESV_1	0h	Always read as 0. Writes have no affect.	Priority
18-16	PRI	R/W	1h	
15-6	RESV_2	0h	Always read as 0. Writes have no affect.	Max wait count
5-0	MAXWAIT	R/W	10h	

### 5.2.11.50 IDMAARBE Register (Offset = 00020204h) [Reset = 00000010h]

IDMAARBE is shown in [Table 5-1366](#).

Return to the [Summary Table](#).

**Table 5-1366. IDMAARBE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	10h	Max wait count

### 5.2.11.51 SDMAARBE Register (Offset = 00020208h) [Reset = 00000001h]

SDMAARBE is shown in [Table 5-1367](#).

Return to the [Summary Table](#).

**Table 5-1367. SDMAARBE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	1h	Max wait count



### 5.2.11.52 ECFGARBE Register (Offset = 00020210h) [Reset = 00070000h]

ECFGARBE is shown in [Table 5-1368](#).

Return to the [Summary Table](#).

**Table 5-1368. ECFGARBE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESV_2	0h	Always read as 0. Writes have no affect.	Priority
18-16	PRI	R/W	7h	
15-0	RESV_1	0h	Always read as 0. Writes have no affect.	

### 5.2.11.53 ICFGMPFAR Register (Offset = 00020300h) [Reset = 00000000h]

ICFGMPFAR is shown in [Table 5-1369](#).

Return to the [Summary Table](#).

**Table 5-1369. ICFGMPFAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	

### 5.2.11.54 ICFGMPFSR Register (Offset = 00020304h) [Reset = 00000000h]

ICFGMPFSR is shown in [Table 5-1370](#).

Return to the [Summary Table](#).

**Table 5-1370. ICFGMPFSR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-8	FID	R	0h	PrivID of faulting requestor
7	SECE	R	0h	Security Error
6	RESV_2	0h	Always read as 0. Writes have no affect.	
5	SR	R	0h	Fault due to Supervisor Read
4	SW	R	0h	Fault due to Supervisor Write
3	SX	R	0h	Fault due to Supervisor Execute
2	UR	R	0h	Fault due to User Read
1	UW	R	0h	Fault due to User Write
0	UX	R	0h	Fault due to User Execute

### 5.2.11.55 ICFGMPFCR Register (Offset = 00020308h) [Reset = 00000000h]

ICFGMPFCR is shown in [Table 5-1371](#).

Return to the [Summary Table](#).

**Table 5-1371. ICFGMPFCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESV_1	0h	Always read as 0. Writes have no affect.	
0	MPFCLR	W	0h	

### 5.2.11.56 ECFGERR Register (Offset = 00020408h) [Reset = 00000000h]

ECFGERR is shown in [Table 5-1372](#).

Return to the [Summary Table](#).

**Table 5-1372. ECFGERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	ERR	R	0h	Error
28-12	RESV_2	0h	Always read as 0. Writes have no affect.	XID
11-8	XID	R	0h	
7-3	RESV_1	0h	Always read as 0. Writes have no affect.	
2-0	STAT	R	0h	Status

### 5.2.11.57 ECFGERRCLR Register (Offset = 0002040Ch) [Reset = 00000000h]

ECFGERRCLR is shown in [Table 5-1373](#).

Return to the [Summary Table](#).

**Table 5-1373. ECFGERRCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESV_1	0h	Always read as 0. Writes have no affect.	
0	CLR	R	0h	Error clear

### 5.2.11.58 PAMAP0 Register (Offset = 00020500h) [Reset = 00000000h]

PAMAP0 is shown in [Table 5-1374](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1374. PAMAP0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	0h	

### 5.2.11.59 PAMAP1 Register (Offset = 00020504h) [Reset = 00000001h]

PAMAP1 is shown in [Table 5-1375](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1375. PAMAP1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	1h	



### 5.2.11.60 PAMAP2 Register (Offset = 00020508h) [Reset = 00000002h]

PAMAP2 is shown in [Table 5-1376](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1376. PAMAP2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	2h	

### 5.2.11.61 PAMAP3 Register (Offset = 0002050Ch) [Reset = 00000003h]

PAMAP3 is shown in [Table 5-1377](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1377. PAMAP3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	3h	

### 5.2.11.62 PAMAP4 Register (Offset = 00020510h) [Reset = 00000004h]

PAMAP4 is shown in [Table 5-1378](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1378. PAMAP4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	4h	

### 5.2.11.63 PAMAP5 Register (Offset = 00020514h) [Reset = 00000005h]

PAMAP5 is shown in [Table 5-1379](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1379. PAMAP5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	5h	

### 5.2.11.64 PAMAP6 Register (Offset = 00020518h) [Reset = 00000006h]

PAMAP6 is shown in [Table 5-1380](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1380. PAMAP6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	6h	

### 5.2.11.65 PAMAP7 Register (Offset = 0002051Ch) [Reset = 00000007h]

PAMAP7 is shown in [Table 5-1381](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1381. PAMAP7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.66 PAMAP8 Register (Offset = 00020520h) [Reset = 00000007h]

PAMAP8 is shown in [Table 5-1382](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1382. PAMAP8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.67 PAMAP9 Register (Offset = 00020524h) [Reset = 00000007h]

PAMAP9 is shown in [Table 5-1383](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1383. PAMAP9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	



### 5.2.11.68 PAMAP10 Register (Offset = 00020528h) [Reset = 00000007h]

PAMAP10 is shown in [Table 5-1384](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1384. PAMAP10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.69 PAMAP11 Register (Offset = 0002052Ch) [Reset = 00000007h]

PAMAP11 is shown in [Table 5-1385](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1385. PAMAP11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.70 PAMAP12 Register (Offset = 00020530h) [Reset = 00000007h]

PAMAP12 is shown in [Table 5-1386](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1386. PAMAP12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.71 PAMAP13 Register (Offset = 00020534h) [Reset = 00000007h]

PAMAP13 is shown in [Table 5-1387](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1387. PAMAP13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.72 PAMAP14 Register (Offset = 00020538h) [Reset = 00000007h]

PAMAP14 is shown in [Table 5-1388](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1388. PAMAP14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.73 PAMAP15 Register (Offset = 0002053Ch) [Reset = 00000007h]

PAMAP15 is shown in [Table 5-1389](#).

Return to the [Summary Table](#).

PAMAP register

**Table 5-1389. PAMAP15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	_RESV1	0h	Always read as 0. Writes have no affect.	AID
2-0	AID	R/W	7h	

### 5.2.11.74 EDCINTMASK Register (Offset = 00031100h) [Reset = 0000007Fh]

EDCINTMASK is shown in [Table 5-1390](#).

Return to the [Summary Table](#).

This read write registers contains the masks for the exceptions generated by each of the EDC error detection units added in this change. Also contained in this register are the bit fields indicating which module has issued an exception.

Reads of this register do not affect the mask bits but do clear the exception bits.

**Table 5-1390. EDCINTMASK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	L1PTAG	R/W	1h	L1PTAG Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.
5	L2LRU	R/W	1h	L2LRU Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.
4	L2SNOP	R/W	1h	L2SNOP Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.
3	L2TAG	R/W	1h	L2TAG Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.
2	L2MPPA	R/W	1h	L2DMPPA Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.
1	L1DTAG	R/W	1h	L1DTAG Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.
0	L1DDATA	R/W	1h	L1DDATA Exception Enable: 1 enables the propagation of the exceptions generated from the error detector. Write 0 to disable exceptions.

### 5.2.11.75 EDCINTFLG Register (Offset = 00031104h) [Reset = 00000000h]

EDCINTFLG is shown in [Table 5-1391](#).

Return to the [Summary Table](#).

This register loads the interrupt flags upon issue by the EDC detection module. These flags are loaded independently of any other flags. Upon a read operation all flags are cleared.

**Table 5-1391. EDCINTFLG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	Reserved
6	L1PTAG	R/W	0h	L1P TAG Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.
5	L2LRU	R/W	0h	L2 LRU Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.
4	L2MPPA	R/W	0h	L2 MPPA Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.
3	L2SNOP	R/W	0h	L2 SNOP Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.
2	L2TAG	R/W	0h	L2TAG Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.
1	L1DTAG	R/W	0h	L1DTAG Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.
0	L1DDATA	R/W	0h	L1DDATA Exception: 1 indicates that an exception has been issued from this error detector. Read of register clears bit.



### 5.2.11.76 L1DEDCMD Register (Offset = 00031108h) [Reset = 00000003h]

L1DEDCMD is shown in [Table 5-1392](#).

Return to the [Summary Table](#).

This write only register provides control for enabling and disabling the ECC logic of the DMC. This register controls the parameters which are accumulated for the Data and Tag RAMs in the DMC. Enabling the ECC allows the counting of recoverable and non-recoverable errors as well as registering the requestor and the address and cache states for any ECC errors which are reported.

**Table 5-1392. L1DEDCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	CCLRTAG	R/W	0h	Correctable Error Clear: Writing 1 clears the correctable error count in the DMCEDPECNT register. This is a one-shot register, and is cleared immediately.
24	CCLRDATA	R/W	0h	Non-Correctable Error Clear: Writing 1 clears the noncorrectable error count in the DMCEDPECNT register. This is a one-shot register and is cleared immediately.
23-18	RESERVED	R	0h	Reserved
17	NCCLTAG	R/W	0h	Correctable Error Clear: Writing 1 clears the correctable error count in the DMCEDPECNT register. This is a one-shot register, and is cleared immediately.
16	NCCLRDATA	R/W	0h	Non-Correctable Error Clear: Writing 1 clears the noncorrectable error count in the DMCEDPECNT register. This is a one-shot register and is cleared immediately.
15-9	RESERVED	R	0h	Reserved
8	SUSPDATA	R/W	0h	Suspend TAG: Write 1 to prevent hamming codes from being written into memory but allows data to be written.
7-2	RESERVED	R	0h	Reserved
1	ENTAG	R/W	1h	Enable Data: Initialized to the active state (1), to disable EDC on Data write 0.
0	ENDATA	R/W	1h	Enable TAG: Initialized to the active state (1), to disable EDC on TAG write 0.

### 5.2.11.77 L1DDCSTAT Register (Offset = 0003110Ch) [Reset = 00000001h]

L1DDCSTAT is shown in [Table 5-1393](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located. . The error information is written upon the occurrence of the first ECC error encountered. CERR1 is set, and the values are retained. If any additional ECC errors occur, this is indicated by setting CERR2.

**Table 5-1393. L1DDCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CERR1	R	0h	Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L1DEDCMD.CCLRDATA
30	CERR2	R	0h	Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L1DEDCMD.CCLRDATA
29-10	RESERVED	R	0h	Reserved
9-7	BANK	R	0h	BANK Number: This value indicates the bank number of the error indicated when CERR1 is 1. This value is the lowest numbered bank if multiple banks error simultaneously. This value is set upon the first error occurring and is cleared by writing a 1 to L1DEDCMD.CCLRDATA.
6-5	BYTE	R	0h	Byte Number: This value indicates the byte number of the error indicated when CERR1 is 1. This value is the lowest numbered byte if multiple bytes error simultaneously in the same bank.This value is set upon the first error occurring and is cleared by writing a 1 to L1DEDCMD.CCLRDATA.
4-2	BITPOS	R	0h	Bit Position: This value indicates the bit position of the error indicated when CERR1 is 1. This value is the specific single error bits from the byte indicated by BYTE and the bank indicated by BANK. This value is set upon the first error occurring and is cleared by writing a 1 to L1DEDCMD.CCLRDATA.
1	SUSP	R	0h	Suspend: EDC logic is in suspend mode, this value is representative of the state of the last write to L1DEDCMD.SUSPDATA.
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENDATA.

### 5.2.11.78 L1DDNCSTAT Register (Offset = 00031110h) [Reset = 00000001h]

L1DDNCSTAT is shown in [Table 5-1394](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located. This register is similar to Error! Not a valid bookmark self-reference. with the exception of the bit position register fields, with two bit errors it is not possible to provide the locations. . The error information is written upon the occurrence of the first ECC error encountered. CERR1 is set, and the values are retained. If any additional ECC errors occur, this is indicated by setting CERR2.

Note: Non-correctable errors do not mean two bits in the 256 bit data word, but two bits+ of error in a single byte. The EDC is capable of correcting up to 8 bits in a single word if there are only single bit errors in each byte.

**Table 5-1394. L1DDNCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR1	R	0h	Non-Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L1DEDCMD.NCCLRDATA
30	NCERR2	R	0h	Non-Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L1DEDCMD.NCCLRDATA
29-10	RESERVED	R	0h	Reserved
9-7	BANK	R	0h	BANK Number: This value indicates the bank number of the error indicated when NCERR1 is 1. This value is the lowest numbered bank if multiple banks error simultaneously. This value is set upon the first error occurring and is cleared by writing a 1 to L1DEDCMD.NCCLRDATA.
6-5	BYTE	R	0h	Byte Number: This value indicates the byte number of the error indicated when NCERR1 is 1. This value is the lowest numbered byte if multiple bytes error simultaneously in the same bank.This value is set upon the first error occurring and is cleared by writing a 1 to L1DEDCMD.NCCLRDATA.
4-2	RESERVED	R	0h	Reserved
1	SUSP	R	0h	Suspend: EDC logic is in suspend mode, this value is representative of the state of the last write to L1DEDCMD.SUSPDATA
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENDATA.

### 5.2.11.79 L1DTCSTAT Register (Offset = 00031114h) [Reset = 00000001h]

L1DTCSTAT is shown in [Table 5-1395](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located. The error information is written upon the occurrence of the first ECC error encountered. CERR1 is set, and the values are retained. If any additional ECC errors occur, this is indicated by setting CERR2.

**Table 5-1395. L1DTCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CERR1	R	0h	Correctable Error 1: Indicates that one correctable error has occurred and that they BANK and BITPOS values are valid. Cleared upon writing a 1 to L1DEDCMD.CCLRDTAG
30	CERR2	R	0h	Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L1DEDCMD.CCLRRTAG
29-9	RESERVED	R	0h	Reserved
8	BANK	R	0h	Bank: Value indicates which bank the correctable error occurred in.
7	CHANNEL	R	0h	Channel: Access to the banks of the L1D TAG memory are split into channels: 1 = [ 53: 43],[ 31: 16] 2 = [ 42: 32],[ 15: 0] Bit Position is an index in the respective channels, this will yield the error bit position.
6-2	BITPOS	R	0h	Bit Position: This value indicates the bit position of the error indicated when CERR1 is 1. This value is the specific single error bits from the byte indicated by BYTE and the bank indicated by BANK. This value is set upon the first error occurring Cleared upon writing a 1 to L1DEDCMD.CCLRRTAG
1	SUSP	R	0h	Suspend: EDC logic is in suspend mode, this value is representative of the state of the last write to L1DEDCMD.SUSPDATA
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENDATA.

### 5.2.11.80 L1DTNCSTAT Register (Offset = 00031118h) [Reset = 00000001h]

L1DTNCSTAT is shown in [Table 5-1396](#).

Return to the [Summary Table](#).

L1D Error Detect TAG Non-Correctable Status Register

**Table 5-1396. L1DTNCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR1	R	0h	Non-Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L1DEDCMD.NCCLRTAG
30	NCERR2	R	0h	Non-Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L1DEDCMD.NCCLRTAG
29-8	RESERVED	R	0h	Reserved
7	BANK	R	0h	BANK Number: This value indicates the bank number of the error indicated when NCERR1 is 1. This value is the lowest numbered bank if multiple banks error simultaneously. This value is set upon the first error occurring and is cleared by writing a 1 to L1DEDCMD.NCCLRDATA.
6-2	RESERVED	R	0h	Reserved
1	SUSP	R	0h	Suspend: EDC logic is in suspend mode, this value is representative of the state of the last write to L1DEDCMD.SUSPDATA
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENDATA.

### 5.2.11.81 L1DDEDADDR Register (Offset = 0003111Ch) [Reset = 00000000h]

L1DDEDADDR is shown in [Table 5-1397](#).

Return to the [Summary Table](#).

The L1D Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in the STATUS registers. The addresses for both correctable and noncorrectable errors are held in this register. The value contained in the addresss field is the lower 16 bits of the word boundary of the memory access. If the access is to 0x00F00020, the address field will read 0x0020, if the access is to 0x00F07FE0, the address field will read 0x7FE0.

**Table 5-1397. L1DDEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L1DEDDNCSTATE.NCERR1, when that field is written to a 1, the address of the noncorrectable error is written to this register field. . Cleared upon writing a 1 to L1DEDCMD.NCCLRDATA
15-11	RESERVED	R	0h	Reserved
10-0	CADDR	R	0h	Correctable Address: The value in this field is qualified by L1DEDDNCSTATE.CERR1, when that field is written to a 1, the address of the non-correctable error is written to this register field. . Cleared upon writing a 1 to L1DEDCMD.NCCLRDATA

### 5.2.11.82 L1DTEADDR Register (Offset = 00031120h) [Reset = 00000000h]

L1DTEADDR is shown in [Table 5-1398](#).

Return to the [Summary Table](#).

The L1D Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in the STATUS registers. The address for both correctable and noncorrectable errors are held in this register. The value of the address is the index (or address on the RAM) of the RAM module.

**Table 5-1398. L1DTEADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L1DEDDNCSTATE.NCERR1, when that field is written to a 1, the address of the noncorrectable error is written to this register field. . Cleared upon writing a 1 to L1DEDCMD.NCCLRDATA
15-11	RESERVED	R	0h	Reserved
10-0	CADDR	R	0h	Correctable Address: The value in this field is qualified by L1DEDDNCSTATE.CERR1, when that field is written to a 1, the address of the non-correctable error is written to this register field. . Cleared upon writing a 1 to L1DEDCMD.NCCLRDATA

### 5.2.11.83 L1DEDCNT Register (Offset = 00031124h) [Reset = 00000000h]

L1DEDCNT is shown in [Table 5-1399](#).

Return to the [Summary Table](#).

The L1D TAG Error Detection Error Count Register stores the error counts for correctable and noncorrectable errors which occur in the L1D Cache Memories, these counts are accumulated regardless of the exception mapping in the L1DEDCMD (0x0182\_1108) interrupt enable bits. The counters capture the total number of correctable and non-correctable errors from both the DATA and TAG memories.

**Table 5-1399. L1DEDCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NCPECNT	R	0h	Non-correctable EDC error count. Shows the number of non-correctable errors which have occurred in the DMC memories since last reset. This counter is cleared by writing a zero value to the counter.
15-0	CPECNT	R	0h	Correctable EDC error count: Shows the number of correctable errors which have occurred in the DMC memories since last reset. This counter is cleared by writing a zero value to the counter.



### 5.2.11.84 L2TEDCMD Register (Offset = 00031128h) [Reset = 000000Fh]

L2TEDCMD is shown in [Table 5-1400](#).

Return to the [Summary Table](#).

This write only register provides control for enabling and disabling the ECC logic of the UMC. This register controls the parameters which are accumulated for the Data and Tag RAMs in the DMC. Enabling the ECC allows the counting of recoverable and non-recoverable errors as well as registering the requestor and the address and cache states for any ECC errors which are reported.

**Table 5-1400. L2TEDCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	Reserved
19	CCLRSNOP	W	0h	Correctable Error Clear, SNOP: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
18	CCLRLRU	W	0h	Correctable Error Clear, LRU: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
17	CCLRMPPA	W	0h	Correctable Error Clear, MPPA: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
16	CCLRTAG	W	0h	Correctable Error Clear, TAG: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
15-12	RESERVED	R	0h	Reserved
11	NCCLRSNOP	W	0h	Non- Correctable Error Clear, SNOP: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
10	NCCLRLRU	W	0h	Non- Correctable Error Clear, LRU: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
9	NCCLRMPPA	W	0h	Non- Correctable Error Clear, MPPA: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
8	NCCLRTAG	W	0h	Non- Correctable Error Clear, TAG: writing 1 to this register cause the corresponding counter to be cleared. The value written to this register is a one shot event.
7-4	RESERVED	R	0h	Reserved
3	SNPEDCEN	W	1h	Enable EDC: writing a 1 to this register enables EDC on the SNOOP Memory
2	LRUEDCEN	W	1h	Enable EDC: writing a 1 to this register enables EDC on the LRU Memory
1	MPPAEDCEN	W	1h	Enable EDC: writing a 1 to this register enables EDC on the MPPA Memory
0	TAGEDCEN	W	1h	Enable EDC: writing a 1 to this register enables the EDC modules for this memory. This bit is persistent and must be actively written to 0 to disable the EDC modules.

### 5.2.11.85 L2TCSTAT Register (Offset = 0003112Ch) [Reset = 00000001h]

L2TCSTAT is shown in [Table 5-1401](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1401. L2TCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CERR1	R	0h	Correctable Error 1: Indicates that one correctable error has occurred and that they BANK and BITPOS values are valid. Cleared upon writing a 1 to L2EDCMD.CCLRDTAG
30	CERR2	R	0h	Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L2EDCMD.CCLRDTAG
29-9	RESERVED	R	0h	Reserved
8	BANK	R	0h	Bank: indicates which of the two banks ( 0 = BANK 0, 1 = BANK 1) contains the correctable error. Cleared upon writing a 1 to L2EDCMD.CCLRDTAG
7	CHANNEL	R	0h	Channel: Access to a bank are split, therefore an indication of channel 1 = [ 41: 21] 2 = [ 20: 0] Bit Position is qualified from these as displacements within the channel.
6-2	BITPOS	R	0h	Bit Position: This value indicates the bit position of the error indicated when CERR1 is 1. This value is the specific single error bits from the byte indicated by BYTE and the bank indicated by BANK. This value is set upon the first error occurring Cleared upon writing a 1 to L2EDCMD.CCLRDTAG
1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENDATA.

### 5.2.11.86 L2TNCSTAT Register (Offset = 00031130h) [Reset = 00000001h]

L2TNCSTAT is shown in [Table 5-1402](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1402. L2TNCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR1	R	0h	Non-Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L1DEDCMD.NCCLRTAG
30	NCERR2	R	0h	Non-Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L1DEDCMD.NCCLRTAG
29-9	RESERVED	R	0h	Reserved
8	BANK	R	0h	Bank: indicates which of the two banks (0 = BANK 0, 1 = BANK 1) contains the correctable error. Cleared upon writing a 1 to L2EDCMD.CCLRTAG
7-1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENDATA.

### 5.2.11.87 L2TEDADDR Register (Offset = 00031134h) [Reset = 00000000h]

L2TEDADDR is shown in [Table 5-1403](#).

Return to the [Summary Table](#).

The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions. The address in these fields is the displacement into the memory as presented to the address port of the SRAM.

**Table 5-1403. L2TEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L2EDDNCSTATE.NCERR1, when that field is written to a 1, the address of the noncorrectable error is written to this register field. . Cleared upon writing a 1 to L2EDCMD.NCCLRDATA
15-11	RESERVED	R	0h	Reserved
10-0	CADDR	R	0h	Correctable Address: The value in this field is qualified by L2EDDNCSTATE.CERR1, when that field is written to a 1, the address of the non-correctable error is written to this register field. . Cleared upon writing a 1 to L2EDCMD.NCCLRDATA

### 5.2.11.88 L2MNCSTAT Register (Offset = 0003113Ch) [Reset = 00000001h]

L2MNCSTAT is shown in [Table 5-1404](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1404. L2MNCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR1	R	0h	Non-Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L2EDCMD.NCCLRMPPA
30	NCERR2	R	0h	Non-Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L2EDCMD.NCCLRMPPA
29-1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L1DEDCMD.ENEDC

### 5.2.11.89 L2MEDADDR Register (Offset = 00031140h) [Reset = 00000000h]

L2MEDADDR is shown in [Table 5-1405](#).

Return to the [Summary Table](#).

The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions.

**Table 5-1405. L2MEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L2EDSNCSTATE.NCERR1, when that field is written to a 1, the address of the non-correctable error is written to this register field.  Cleared upon writing a 1 to L2EDCMD.NCCLRMPPA
15-0	RESERVED	R	0h	Reserved

### 5.2.11.90 L2SCSTAT Register (Offset = 00031144h) [Reset = 00000001h]

L2SCSTAT is shown in [Table 5-1406](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1406. L2SCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CERR1	R	0h	Correctable Error 1: Indicates that one correctable error has occurred and that they BANK and BITPOS values are valid. Cleared upon writing a 1 to L2EDCMD.CCLRDSNOP
30	CERR2	R	0h	Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L2EDCMD.CCLRSNOP
29-7	RESERVED	R	0h	Reserved
6-2	BITPOS	R	0h	Bit Position: This value indicates the bit position of the error indicated when CERR1 is 1. This value is the specific single error bits from the byte indicated by BYTE and the bank indicated by BANK. This value is set upon the first error occurring Cleared upon writing a 1 to L2EDCMD.CCLRSNOP
1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L2EDCMD.ENEDC

### 5.2.11.91 L2SNCSTAT Register (Offset = 00031148h) [Reset = 00000001h]

L2SNCSTAT is shown in [Table 5-1407](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1407. L2SNCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR1	R	0h	Non-Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L2EDCMD.NCCLRSNOP
30	NCERR2	R	0h	Non-Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L2EDCMD.NCCLRSNOP
29-1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L2EDCMD.ENEDC



### 5.2.11.92 L2SEDADDR Register (Offset = 0003114Ch) [Reset = 00000000h]

L2SEDADDR is shown in [Table 5-1408](#).

Return to the [Summary Table](#).

The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions.

**Table 5-1408. L2SEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-26	RESERVED	R	0h	Reserved
25-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L2EDDNCSTATE.NCERR1, when that field is written to a 1, the address of the noncorrectable error is written to this register field. . Cleared upon writing a 1 to L2EDCMD.NCCLRSNOP
15-10	RESERVED	R	0h	Reserved
9-0	CADDR	R	0h	Correctable Address: The value in this field is qualified by L2EDDNCSTATE.CERR1, when that field is written to a 1, the address of the non-correctable error is written to this register field. . Cleared upon writing a 1 to L2EDCMD.NCCLRSNOP

### 5.2.11.93 L2LCSTAT Register (Offset = 00031150h) [Reset = 00000001h]

L2LCSTAT is shown in [Table 5-1409](#).

Return to the [Summary Table](#).

This read only register contains the current error data for Correctable ECC errors. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1409. L2LCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	CERR1	R	0h	Correctable Error 1: Indicates that one correctable error has occurred and that they BANK and BITPOS values are valid. Cleared upon writing a 1 to L2EDCMD.CCLRDLRU
30	CERR2	R	0h	Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L2EDCMD.CCLRRLRU
29-7	RESERVED	R	0h	Reserved
6-2	BITPOS	R	0h	Bit Position: This value indicates the bit position of the error indicated when CERR1 is 1. This value is the specific single error bits from the byte indicated by BYTE and the bank indicated by BANK. This value is set upon the first error occurring Cleared upon writing a 1 to L2EDCMD.CCLRRLRU
1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L2EDCMD.ENEDC

### 5.2.11.94 L2LNCSTAT Register (Offset = 00031154h) [Reset = 00000001h]

L2LNCSTAT is shown in [Table 5-1410](#).

Return to the [Summary Table](#).

**Table 5-1410. L2LNCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR1	R	0h	Non-Correctable Error 1: Indicates that one correctable error has occurred and that they BANK, BYTE and BITPOS values are valid. Cleared upon writing a 1 to L2EDCMD.NCCLR:LRU
30	NCERR2	R	0h	Non-Correctable Error 2: Indicates that a subsequent correctable error has occurred ,BANK, BYTE and BITPOS values are valid but apply to the error reported in CERR1. Cleared upon writing a 1 to L2EDCMD.NCCLRLRU
29-1	RESERVED	R	0h	Reserved
0	EN	R	1h	EDC Enable: Indicates EDC logic is enabled, this value is representative of the state of the last write to L2EDCMD.ENEDC

### 5.2.11.95 L2LEDADDR Register (Offset = 00031158h) [Reset = 00000000h]

L2LEDADDR is shown in [Table 5-1411](#).

Return to the [Summary Table](#).

The L2Correctable Error Detection Address Register pair contain the additional information on correctable errors reported in 1.3.1.12. With both exceptions activated, the TAG exception will take priority on simultaneous exceptions.

**Table 5-1411. L2LEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L2EDDNCSTATE.NCERR1, when that field is written to a 1, the address of the noncorrectable error is written to this register field. . Cleared upon writing a 1 to L2EDCMD.NCCLRLRU
15-10	RESERVED	R	0h	
9-0	CADDR	R	0h	Correctable Address: The value in this field is qualified by L2EDDNCSTATE.CERR1, when that field is written to a 1, the address of the non-correctable error is written to this register field. . Cleared upon writing a 1 to L2EDCMD.NCCLRLRU

### 5.2.11.96 L2TEDCNT Register (Offset = 0003115Ch) [Reset = 00000000h]

L2TEDCNT is shown in [Table 5-1412](#).

Return to the [Summary Table](#).

The L2 Error Detection Error Count Register stores the error counts for correctable and non-correctable errors which occur in the L1D Data Memory, these counts are accumulated regardless of the exception mapping in the L1DEDCMD (0x0182\_1108) interrupt enable bits. The counts contained in these registers are the accumulation of all L2 errors. This value saturates at an unsigned 0xFFFF.

**Table 5-1412. L2TEDCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NCPECNT	R	0h	Non-correctable EDC error count. Shows the number of non-correctable errors which have occurred in the DMC memories since last reset. This counter is cleared by writing a zero value to the counter. CPECNT 15-0 0 Correctable EDC error count:
15-0	CPECNT	R	0h	Correctable EDC error count: Shows the number of correctable errors which have occurred in the DMC memories since last reset. This counter is cleared by writing a zero value to the counter.

### 5.2.11.97 L1PTEDCMD Register (Offset = 00031160h) [Reset = 00000001h]

L1PTEDCMD is shown in [Table 5-1413](#).

Return to the [Summary Table](#).

This write only register provides control for enabling and disabling the Parity logic of the PMC. This register controls the parameters which are accumulated for the Data and Tag RAMs in the PMC. Enabling the Parity allows the counting of recoverable and non-recoverable errors as well as registering the requestor and the address and cache states for any Parity errors which are reported.

**Table 5-1413. L1PTEDCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	Reserved
5	NCCLR	W	0h	Non-Correctable Error Clear: Clears the non-correctable error count in the DMCEDPECNT register
4-1	RESERVED	R	0h	Reserved
0	EN	W	1h	Enable Parity: Enables the Parity logic (parity_enable)

### 5.2.11.98 L1PTEDSTAT Register (Offset = 00031164h) [Reset = 00000001h]

L1PTEDSTAT is shown in [Table 5-1414](#).

Return to the [Summary Table](#).

This read only register contains the current error. The register provides information on whether the error was correctable or non-correctable, the requestor, and what position the error was located.

**Table 5-1414. L1PTEDSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NCERR2	R	0h	Non-Correctable Error: Indicates that there was a seconde non-correctable error, Also indicates the logged error is valid.
30	NCERR1	R	0h	Non-Correctable Error: Indicates the first error was noncorrectable.
29-1	RESERVED	R	0h	Reserved
0	EN	R	1h	Enable: reflects the current state of the EN bit from L1PEDCMD

### 5.2.11.99 L1PTEDADDR Register (Offset = 00031168h) [Reset = 00000000h]

L1PTEDADDR is shown in [Table 5-1415](#).

Return to the [Summary Table](#).

The address error registers for correctable and non-correctable errors are identical in structure, the indication applies to the correctable and non-correctable errors, they are both covered in this section. The register provides a record of the address of the detected error and the access status or the cache.

**Table 5-1415. L1PTEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26-16	NCADDR	R	0h	Non-Correctable Address: The value in this field is qualified by L1PEDDNCSTATE.NCERR, when that field is written to a 1, the address of the non-correctable error is written to this register field.  Cleared upon writing a 1 to L1PEDCMD.NCCLRLR
15-0	RESERVED	R	0h	Reserved



### 5.2.11.100 L1DTEDCNT Register (Offset = 0003116Ch) [Reset = 00000000h]

L1DTEDCNT is shown in [Table 5-1416](#).

Return to the [Summary Table](#).

This read only register contains the counts of correctable and non-correctable errors which have occurred. To reset these counts write to the DMCEDCMD register in the appropriate bit.

**Table 5-1416. L1DTEDCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NCPECNT	R	0h	Non-correctable EDC error count. Shows the number of non-correctable errors which have occurred in the DMC memories since last reset. This counter is cleared by writing a zero value to the counter.
15-0	RESERVED	R	0h	Reserved

### 5.2.11.101 L2CFG Register (Offset = 00040000h) [Reset = 01000000h]

L2CFG is shown in [Table 5-1417](#).

Return to the [Summary Table](#).

**Table 5-1417. L2CFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESV_1	0h	Always read as 0. Writes have no affect.	
27-24	NUM_MM	R	1h	
23-20	RESV_2	0h	Always read as 0. Writes have no affect.	
19-16	MMID	R	0h	
15-10	RESV_3	0h	Always read as 0. Writes have no affect.	
9	IP	0h		
8	ID	0h		
7-5	RESV_4	0h	Always read as 0. Writes have no affect.	
4-3	L2CC	0h		
2-0	L2MODE	R/W	0h	

### 5.2.11.102 L1PCFG Register (Offset = 00040020h) [Reset = 00000007h]

L1PCFG is shown in [Table 5-1418](#).

Return to the [Summary Table](#).

**Table 5-1418. L1PCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESV_1	0h	Always read as 0. Writes have no affect.	
2-0	L1PMODE	R/W	7h	

### 5.2.11.103 L1PCC Register (Offset = 00040024h) [Reset = 00000000h]

L1PCC is shown in [Table 5-1419](#).

Return to the [Summary Table](#).

**Table 5-1419. L1PCC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESV_1	0h	Always read as 0. Writes have no affect.	
18-16	POPER	R	0h	
15-3	RESV_2	0h	Always read as 0. Writes have no affect.	
2-0	OPER	R/W	0h	

### 5.2.11.104 L1DCFG Register (Offset = 00040040h) [Reset = 00000007h]

L1DCFG is shown in [Table 5-1420](#).

Return to the [Summary Table](#).

**Table 5-1420. L1DCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESV_1	0h	Always read as 0. Writes have no affect.	
2-0	L1DMODE	R/W	7h	

### 5.2.11.105 L1DCC Register (Offset = 00040044h) [Reset = 00000000h]

L1DCC is shown in [Table 5-1421](#).

Return to the [Summary Table](#).

**Table 5-1421. L1DCC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESV_1	0h	Always read as 0. Writes have no affect.	
18-16	POPER	R	0h	
15-3	RESV_2	0h	Always read as 0. Writes have no affect.	
2-0	OPER	R/W	0h	

### 5.2.11.106 CPUARBU Register (Offset = 00041000h) [Reset = 00010010h]

CPUARBU is shown in [Table 5-1422](#).

Return to the [Summary Table](#).

**Table 5-1422. CPUARBU Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESV_1	0h	Always read as 0. Writes have no affect.	Priority
18-16	PRI	R/W	1h	
15-6	RESV_2	0h	Always read as 0. Writes have no affect.	Max wait count
5-0	MAXWAIT	R/W	10h	

### 5.2.11.107 IDMAARBU Register (Offset = 00041004h) [Reset = 00000010h]

IDMAARBU is shown in [Table 5-1423](#).

Return to the [Summary Table](#).

**Table 5-1423. IDMAARBU Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	10h	Max wait count



### 5.2.11.108 SDMAARBU Register (Offset = 00041008h) [Reset = 00000001h]

SDMAARBU is shown in [Table 5-1424](#).

Return to the [Summary Table](#).

**Table 5-1424. SDMAARBU Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	1h	Max wait count

### 5.2.11.109 UCARBU Register (Offset = 0004100Ch) [Reset = 00000020h]

UCARBU is shown in [Table 5-1425](#).

Return to the [Summary Table](#).

**Table 5-1425. UCARBU Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	20h	Max wait count

### 5.2.11.110 MDMAARBU Register (Offset = 00041010h) [Reset = 06070000h]

MDMAARBU is shown in [Table 5-1426](#).

Return to the [Summary Table](#).

**Table 5-1426. MDMAARBU Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESV_1	0h	Always read as 0. Writes have no affect.	Elevated priority
26-24	UPRI	R/W	6h	
23-19	RESV_2	0h	Always read as 0. Writes have no affect.	Default priority
18-16	PRI	R/W	7h	
15-0	RESV_3	0h	Always read as 0. Writes have no affect.	

### 5.2.11.111 CPUARBD Register (Offset = 00041040h) [Reset = 00010010h]

CPUARBD is shown in [Table 5-1427](#).

Return to the [Summary Table](#).

**Table 5-1427. CPUARBD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESV_1	0h	Always read as 0. Writes have no affect.	Priority
18-16	PRI	R/W	1h	
15-6	RESV_2	0h	Always read as 0. Writes have no affect.	Max wait count
5-0	MAXWAIT	R/W	10h	

### 5.2.11.112 IDMAARBD Register (Offset = 00041044h) [Reset = 00000010h]

IDMAARBD is shown in [Table 5-1428](#).

Return to the [Summary Table](#).

**Table 5-1428. IDMAARBD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	10h	Max wait count

### 5.2.11.113 SDMAARBD Register (Offset = 00041048h) [Reset = 00000001h]

SDMAARBD is shown in [Table 5-1429](#).

Return to the [Summary Table](#).

**Table 5-1429. SDMAARBD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	1h	Max wait count

### 5.2.11.114 UCARBD Register (Offset = 0004104Ch) [Reset = 00000020h]

UCARBD is shown in [Table 5-1430](#).

Return to the [Summary Table](#).

**Table 5-1430. UCARBD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESV_1	0h	Always read as 0. Writes have no affect.	
5-0	MAXWAIT	R/W	20h	Max wait count

### 5.2.11.115 L2WBAR Register (Offset = 00044000h) [Reset = 00000000h]

L2WBAR is shown in [Table 5-1431](#).

Return to the [Summary Table](#).

**Table 5-1431. L2WBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register.



### 5.2.11.116 L2WWC Register (Offset = 00044004h) [Reset = 00000000h]

L2WWC is shown in [Table 5-1432](#).

Return to the [Summary Table](#).

**Table 5-1432. L2WWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.117 L2WIBAR Register (Offset = 00044010h) [Reset = 00000000h]

L2WIBAR is shown in [Table 5-1433](#).

Return to the [Summary Table](#).

**Table 5-1433. L2WIBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register.

### 5.2.11.118 L2WIWC Register (Offset = 00044014h) [Reset = 00000000h]

L2WIWC is shown in [Table 5-1434](#).

Return to the [Summary Table](#).

**Table 5-1434. L2WIWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.119 L2IBAR Register (Offset = 00044018h) [Reset = 00000000h]

L2IBAR is shown in [Table 5-1435](#).

Return to the [Summary Table](#).

**Table 5-1435. L2IBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register.

### 5.2.11.120 L2IWC Register (Offset = 0004401Ch) [Reset = 00000000h]

L2IWC is shown in [Table 5-1436](#).

Return to the [Summary Table](#).

**Table 5-1436. L2IWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.121 L1PIBAR Register (Offset = 00044020h) [Reset = 00000000h]

L1PIBAR is shown in [Table 5-1437](#).

Return to the [Summary Table](#).

**Table 5-1437. L1PIBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register.

### 5.2.11.122 L1PIWC Register (Offset = 00044024h) [Reset = 00000000h]

L1PIWC is shown in [Table 5-1438](#).

Return to the [Summary Table](#).

**Table 5-1438. L1PIWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.123 L1DWIBAR Register (Offset = 00044030h) [Reset = 00000000h]

L1DWIBAR is shown in [Table 5-1439](#).

Return to the [Summary Table](#).

**Table 5-1439. L1DWIBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register



### 5.2.11.124 L1DWIWC Register (Offset = 00044034h) [Reset = 00000000h]

L1DWIWC is shown in [Table 5-1440](#).

Return to the [Summary Table](#).

**Table 5-1440. L1DWIWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.125 L1DWBAR Register (Offset = 00044040h) [Reset = 00000000h]

L1DWBAR is shown in [Table 5-1441](#).

Return to the [Summary Table](#).

**Table 5-1441. L1DWBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register

### 5.2.11.126 L1DWWC Register (Offset = 00044044h) [Reset = 00000000h]

L1DWWC is shown in [Table 5-1442](#).

Return to the [Summary Table](#).

**Table 5-1442. L1DWWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.127 L1DIBAR Register (Offset = 00044048h) [Reset = 00000000h]

L1DIBAR is shown in [Table 5-1443](#).

Return to the [Summary Table](#).

**Table 5-1443. L1DIBAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	W	0h	Write only address register

### 5.2.11.128 L1DIWC Register (Offset = 0004404Ch) [Reset = 00000000h]

L1DIWC is shown in [Table 5-1444](#).

Return to the [Summary Table](#).

**Table 5-1444. L1DIWC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-0	WC	R/W	0h	

### 5.2.11.129 L2WB Register (Offset = 00045000h) [Reset = 00000000h]

L2WB is shown in [Table 5-1445](#).

Return to the [Summary Table](#).

**Table 5-1445. L2WB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C	R/W	0h	

### 5.2.11.130 L2WBINV Register (Offset = 00045004h) [Reset = 00000000h]

L2WBINV is shown in [Table 5-1446](#).

Return to the [Summary Table](#).

**Table 5-1446. L2WBINV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C	R/W	0h	

### 5.2.11.131 L2INV Register (Offset = 00045008h) [Reset = 00000000h]

L2INV is shown in [Table 5-1447](#).

Return to the [Summary Table](#).

**Table 5-1447. L2INV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	I	R/W	0h	



### 5.2.11.132 L1PINV Register (Offset = 00045028h) [Reset = 00000000h]

L1PINV is shown in [Table 5-1448](#).

Return to the [Summary Table](#).

**Table 5-1448. L1PINV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	I	R/W	0h	

### 5.2.11.133 L1DWB Register (Offset = 00045040h) [Reset = 00000000h]

L1DWB is shown in [Table 5-1449](#).

Return to the [Summary Table](#).

**Table 5-1449. L1DWB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C	R/W	0h	

### 5.2.11.134 L1DWBINV Register (Offset = 00045044h) [Reset = 00000000h]

L1DWBINV is shown in [Table 5-1450](#).

Return to the [Summary Table](#).

**Table 5-1450. L1DWBINV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C	R/W	0h	

### 5.2.11.135 L1DINV Register (Offset = 00045048h) [Reset = 00000000h]

L1DINV is shown in [Table 5-1451](#).

Return to the [Summary Table](#).

**Table 5-1451. L1DINV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	I	R/W	0h	

### 5.2.11.136 L2EDSTAT Register (Offset = 00046004h) [Reset = 00000001h]

L2EDSTAT is shown in [Table 5-1452](#).

Return to the [Summary Table](#).

L2 EDC Status register

**Table 5-1452. L2EDSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESV1	0h	Always read as 0. Writes have no affect.	Bit position of Error
23-16	BITPOS	R	0h	
15-10	RESV2	0h	Always read as 0. Writes have no affect.	
9-8	NERR	R	0h	Non correctable Error
7	VERR	R	0h	Victim EDC Error
6	DMAERR	R	0h	DMA EDC Error
5	PERR	R	0h	PMC Data EDC Error
4	DERR	R	0h	DMC Data EDC Error
3	SUSP	R	0h	EDC Suspend
2	DIS	R	0h	EDC Disable
1	RESV3	0h	Always read as 0. Writes have no affect.	EDC Enable
0	EN	R	1h	

### 5.2.11.137 L2EDCMD Register (Offset = 00046008h) [Reset = 00000001h]

L2EDCMD is shown in [Table 5-1453](#).

Return to the [Summary Table](#).

L2 EDC Command register

**Table 5-1453. L2EDCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESV1	0h	Always read as 0. Writes have no affect.	
7	VCLR	W	0h	Victim EDC Error
6	DMACLR	W	0h	DMA EDC Error
5	PCLR	W	0h	PMC Data EDC Error
4	DCLR	W	0h	DMC Data EDC Error
3	SUSP	W	0h	EDC Suspend
2	DIS	W	0h	EDC Disable
1	RESV3	0h	Always read as 0. Writes have no affect.	
0	EN	W	1h	EDC Enable

### 5.2.11.138 L2EDADDR Register (Offset = 0004600Ch) [Reset = 00000000h]

L2EDADDR is shown in [Table 5-1454](#).

Return to the [Summary Table](#).

L2 EDC error address

**Table 5-1454. L2EDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	ADDR	R	0h	L2 Error detection address
4-3	WAY	R	0h	Addr way
2-1	RESV1	0h	Always read as 0. Writes have no affect.	
0	RAM	R	0h	Addr is sram

### 5.2.11.139 L2EDCPEC Register (Offset = 00046018h) [Reset = 00000000h]

L2EDCPEC is shown in [Table 5-1455](#).

Return to the [Summary Table](#).

L2 EDC correctable error count

**Table 5-1455. L2EDCPEC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESV1	0h	Always read as 0. Writes have no affect.	count
7-0	CNT	R	0h	



### 5.2.11.140 L2EDCNEC Register (Offset = 0004601Ch) [Reset = 00000000h]

L2EDCNEC is shown in [Table 5-1456](#).

Return to the [Summary Table](#).

L2 EDC noncorrectable error count

**Table 5-1456. L2EDCNEC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESV1	0h	Always read as 0. Writes have no affect.	count
7-0	CNT	R	0h	

### 5.2.11.141 MDMAERR Register (Offset = 00046020h) [Reset = 00000000h]

MDMAERR is shown in [Table 5-1457](#).

Return to the [Summary Table](#).

**Table 5-1457. MDMAERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	ERR	R	0h	Error
28-12	RESV_2	0h	Always read as 0. Writes have no affect.	
11-8	XID	R	0h	XID
7-3	RESV_1	0h	Always read as 0. Writes have no affect.	
2-0	STAT	R	0h	Status

### 5.2.11.142 MDMAERRCLR Register (Offset = 00046024h) [Reset = 00000000h]

MDMAERRCLR is shown in [Table 5-1458](#).

Return to the [Summary Table](#).

**Table 5-1458. MDMAERRCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESV_1	0h	Always read as 0. Writes have no affect.	
0	CLR	R	0h	Error clear

### 5.2.11.143 L2EDCEN Register (Offset = 00046030h) [Reset = 0000001Fh]

L2EDCEN is shown in [Table 5-1459](#).

Return to the [Summary Table](#).

L2 EDC enable register

**Table 5-1459. L2EDCEN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESV1	0h	Always read as 0. Writes have no affect.	
4	SDMAEN	W	1h	Enable edc on sdma reads to l2 sram
3	PL2SEN	W	1h	Enable edc on pmc hits to l2 sram
2	DL2SEN	W	1h	Enable edc on dmc hits to l2 sram
1	PL2CEN	W	1h	Enable edc on pmc hits to l2 cache
0	DL2CEN	W	1h	Enable edc on dmc hits to l2 cache

### 5.2.11.144 L1PEDSTAT Register (Offset = 00046404h) [Reset = 00000000h]

L1PEDSTAT is shown in [Table 5-1460](#).

Return to the [Summary Table](#).

L1P EDC Status register

**Table 5-1460. L1PEDSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESV1	0h	Always read as 0. Writes have no affect.	DMA Error
6	DMAERR	R	0h	
5	PERR	R	0h	
4	RESV2	0h	Always read as 0. Writes have no affect.	EDC Suspend
3	SUSP	R	0h	
2	DIS	R	0h	
1	RESV3	0h	Always read as 0. Writes have no affect.	EDC Disable
0	EN	R	0h	

### 5.2.11.145 L1PEDCMD Register (Offset = 00046408h) [Reset = 00000000h]

L1PEDCMD is shown in [Table 5-1461](#).

Return to the [Summary Table](#).

L1P EDC Command register

**Table 5-1461. L1PEDCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESV1	0h	Always read as 0. Writes have no affect.	
6	DMACLR	W	0h	Clear DMA Error
5	PCLR	W	0h	Clear Parity Error
4	RESV2	0h	Always read as 0. Writes have no affect.	
3	SUSP	W	0h	EDC Suspend
2	DIS	W	0h	EDC Disable
1	RESV3	0h	Always read as 0. Writes have no affect.	
0	EN	W	0h	EDC Enable

### 5.2.11.146 L1PEDADDR Register (Offset = 0004640Ch) [Reset = 00000000h]

L1PEDADDR is shown in [Table 5-1462](#).

Return to the [Summary Table](#).

L1P EDC error address

**Table 5-1462. L1PEDADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	ADDR	R	0h	L1P Error detection address
4-1	RESV1	0h	Always read as 0. Writes have no affect.	Addr is sram
0	RAM	R	0h	

### 5.2.11.147 MAR Register (Offset = 00048000h) [Reset = 00000001h]

MAR is shown in [Table 5-1463](#).

Return to the [Summary Table](#).

MAR register

**Table 5-1463. MAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESV_1	0h	Always read as 0. Writes have no affect.	
3	PFX	R	0h	Hard coded to 0
2	PCX	R	0h	Hard coded to 0
1	WTE	R	0h	Hard coded to 0
0	PC	R	1h	Hard coded to 1



### 5.2.11.148 L2MPFAR Register (Offset = 0004A000h) [Reset = 00000000h]

L2MPFAR is shown in [Table 5-1464](#).

Return to the [Summary Table](#).

**Table 5-1464. L2MPFAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	

### 5.2.11.149 L2MPFSR Register (Offset = 0004A004h) [Reset = 00000000h]

L2MPFSR is shown in [Table 5-1465](#).

Return to the [Summary Table](#).

**Table 5-1465. L2MPFSR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-9	FID	R	0h	PrivID of faulting requestor
8	LOCAL	R	0h	Fault due to local CPU access
7	NS	R	0h	Fault due to non-secure code
6	RESV_2	0h	Always read as 0. Writes have no affect.	
5	SR	R	0h	Fault due to Supervisor Read
4	SW	R	0h	Fault due to Supervisor Write
3	SX	R	0h	Fault due to Supervisor Execute
2	UR	R	0h	Fault due to User Read
1	UW	R	0h	Fault due to User Write
0	UX	R	0h	Fault due to User Execute

### 5.2.11.150 L2MPFCR Register (Offset = 0004A008h) [Reset = 00000000h]

L2MPFCR is shown in [Table 5-1466](#).

Return to the [Summary Table](#).

**Table 5-1466. L2MPFCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESV_1	0h	Always read as 0. Writes have no affect.	
0	MPFCLR	W	0h	

### 5.2.11.151 L2MPPA Register (Offset = 0004A200h) [Reset = 0000FFFFh]

L2MPPA is shown in [Table 5-1467](#).

Return to the [Summary Table](#).

L2 Memory Protection Page Attribute register

**Table 5-1467. L2MPPA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15	AID5	R/W	1h	Allow ID 5 to access page
14	AID4	R/W	1h	Allow ID 4 to access page
13	AID3	R/W	1h	Allow ID 3 to access page
12	AID2	R/W	1h	Allow ID 2 to access page
11	AID1	R/W	1h	Allow ID 1 to access page
10	AID0	R/W	1h	Allow ID 0 to access page
9	AIDX	R/W	1h	Allow IDs > 5 to access page
8	LOCAL	R/W	1h	Allow CPU to access page
7	NS	0h	Non-secure (Secure devices only)	
6	EMU	0h	Emulatable (Secure devices only)	
5	SR	R/W	1h	Supervisor Read Access
4	SW	R/W	1h	Supervisor Write Access
3	SX	R/W	1h	Supervisor Execute Access
2	UR	R/W	1h	User Read Access
1	UW	R/W	1h	User Write Access
0	UX	R/W	1h	User Execute Access

### 5.2.11.152 L1PMPFAR Register (Offset = 0004A400h) [Reset = 00000000h]

L1PMPFAR is shown in [Table 5-1468](#).

Return to the [Summary Table](#).

**Table 5-1468. L1PMPFAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	

### 5.2.11.153 L1PMPFSR Register (Offset = 0004A404h) [Reset = 00000000h]

L1PMPFSR is shown in [Table 5-1469](#).

Return to the [Summary Table](#).

**Table 5-1469. L1PMPFSR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-9	FID	R	0h	PrivID of faulting requestor
8	LOCAL	R	0h	Fault due to local CPU access
7	NS	R	0h	Fault due to non-secure code
6	RESV_2	0h	Always read as 0. Writes have no affect.	
5	SR	R	0h	Fault due to Supervisor Read
4	SW	R	0h	Fault due to Supervisor Write
3	SX	R	0h	Fault due to Supervisor Execute
2	UR	R	0h	Fault due to User Read
1	UW	R	0h	Fault due to User Write
0	UX	R	0h	Fault due to User Execute

### 5.2.11.154 L1PMPFCR Register (Offset = 0004A408h) [Reset = 00000000h]

L1PMPFCR is shown in [Table 5-1470](#).

Return to the [Summary Table](#).

**Table 5-1470. L1PMPFCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESV_1	0h	Always read as 0. Writes have no affect.	
0	MPFCLR	W	0h	

### 5.2.11.155 L1PMPPA Register (Offset = 0004A640h) [Reset = 0000FFFFh]

L1PMPPA is shown in [Table 5-1471](#).

Return to the [Summary Table](#).

L1P Memory Protection Page Attribute register

**Table 5-1471. L1PMPPA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15	AID5	R/W	1h	Allow ID 5 to access page
14	AID4	R/W	1h	Allow ID 4 to access page
13	AID3	R/W	1h	Allow ID 3 to access page
12	AID2	R/W	1h	Allow ID 2 to access page
11	AID1	R/W	1h	Allow ID 1 to access page
10	AID0	R/W	1h	Allow ID 0 to access page
9	AIDX	R/W	1h	Allow IDs > 5 to access page
8	LOCAL	R/W	1h	Allow CPU to access page
7	NS	0h	Non-secure (Secure devices only)	
6	EMU	0h	Emulatable (Secure devices only)	
5	SR	R/W	1h	Supervisor Read Access
4	SW	R/W	1h	Supervisor Write Access
3	SX	R/W	1h	Supervisor Execute Access
2	UR	R/W	1h	User Read Access
1	UW	R/W	1h	User Write Access
0	UX	R/W	1h	User Execute Access



### 5.2.11.156 L1DMPFAR Register (Offset = 0004AC00h) [Reset = 00000000h]

L1DMPFAR is shown in [Table 5-1472](#).

Return to the [Summary Table](#).

**Table 5-1472. L1DMPFAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	

### 5.2.11.157 L1DMPFSR Register (Offset = 0004AC04h) [Reset = 00000000h]

L1DMPFSR is shown in [Table 5-1473](#).

Return to the [Summary Table](#).

**Table 5-1473. L1DMPFSR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15-9	FID	R	0h	PrivID of faulting requestor
8	LOCAL	R	0h	Fault due to local CPU access
7	NS	R	0h	Fault due to non-secure code
6	RESV_2	0h	Always read as 0. Writes have no affect.	
5	SR	R	0h	Fault due to Supervisor Read
4	SW	R	0h	Fault due to Supervisor Write
3	SX	R	0h	Fault due to Supervisor Execute
2	UR	R	0h	Fault due to User Read
1	UW	R	0h	Fault due to User Write
0	UX	R	0h	Fault due to User Execute

### 5.2.11.158 L1DMPFCR Register (Offset = 0004AC08h) [Reset = 00000000h]

L1DMPFCR is shown in [Table 5-1474](#).

Return to the [Summary Table](#).

**Table 5-1474. L1DMPFCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESV_1	0h	Always read as 0. Writes have no affect.	
0	MPFCLR	W	0h	

### 5.2.11.159 MPLK Register (Offset = 0004AD00h) [Reset = 00000000h]

MPLK is shown in [Table 5-1475](#).

Return to the [Summary Table](#).

MPLK register

**Table 5-1475. MPLK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MPLK	W	0h	

### 5.2.11.160 MPLKCMD Register (Offset = 0004AD10h) [Reset = 00000000h]

MPLKCMD is shown in [Table 5-1476](#).

Return to the [Summary Table](#).

**Table 5-1476. MPLKCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESV_1	0h	Always read as 0. Writes have no affect.	
2	KEYR	W	0h	
1	LOCK	W	0h	
0	UNLOCK	W	0h	

### 5.2.11.161 MPLKSTAT Register (Offset = 0004AD14h) [Reset = 00000002h]

MPLKSTAT is shown in [Table 5-1477](#).

Return to the [Summary Table](#).

**Table 5-1477. MPLKSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESV_1	0h	Always read as 0. Writes have no affect.	
1	NSL	R	1h	
0	LK	R	0h	

### 5.2.11.162 L1DMPPA Register (Offset = 0004AE40h) [Reset = 0000FFFFh]

L1DMPPA is shown in [Table 5-1478](#).

Return to the [Summary Table](#).

L1D Memory Protection Page Attribute register

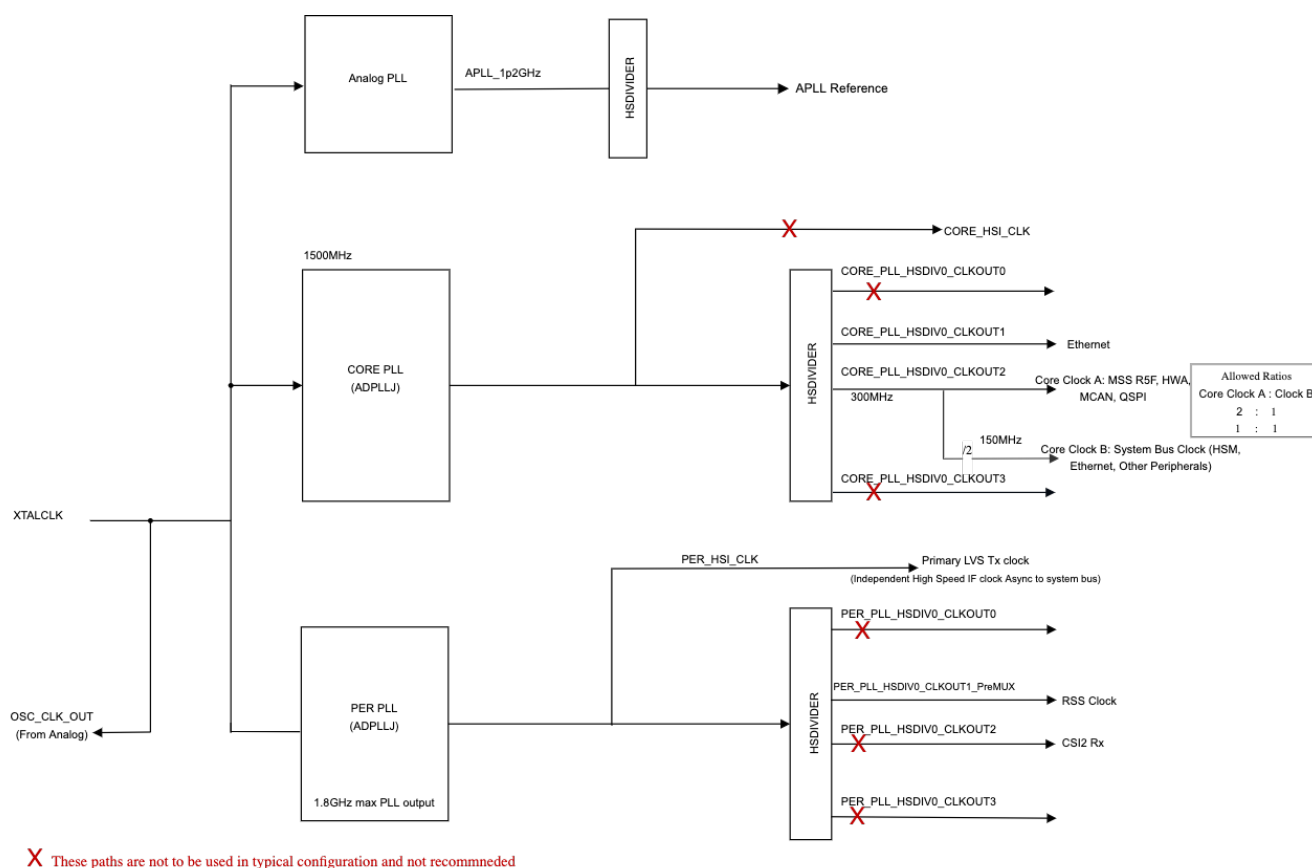
**Table 5-1478. L1DMPPA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESV_1	0h	Always read as 0. Writes have no affect.	
15	AID5	R/W	1h	Allow ID 5 to access page
14	AID4	R/W	1h	Allow ID 4 to access page
13	AID3	R/W	1h	Allow ID 3 to access page
12	AID2	R/W	1h	Allow ID 2 to access page
11	AID1	R/W	1h	Allow ID 1 to access page
10	AID0	R/W	1h	Allow ID 0 to access page
9	AIDX	R/W	1h	Allow IDs > 5 to access page
8	LOCAL	R/W	1h	Allow CPU to access page
7	NS	0h	Non-secure (Secure devices only)	
6	EMU	0h	Emulatable (Secure devices only)	
5	SR	R/W	1h	Supervisor Read Access
4	SW	R/W	1h	Supervisor Write Access
3	SX	R/W	1h	Supervisor Execute Access
2	UR	R/W	1h	User Read Access
1	UW	R/W	1h	User Write Access
0	UX	R/W	1h	User Execute Access

## 5.3 Device Clock Architecture

### 5.3.1 Clock Overview

[Figure 5-2](#) shows a high-level overview of the device clock architecture. The figure captures the key clock sources and the configuration options available to select the appropriate clock source. The detailed structure is captured under each PLL clocking section.



**Figure 5-2. Clock Tree Configuration**

There are 2 clock sources for AWR2544

1. Crystal
2. External Clock source

The 40 MHz and 50 MHz crystals are supported.

The external crystal is connected to an oscillator that generates 40MHz/50MHz clock. This clock is fed to PLL's. There are 3 PLL's

- Analog PLL - APLL
- Digital PLL for Core of type ADPLLJ → Core PLL
- Digital PLL for peripherals of type ADPLLJ → Per PLL

There is a RC Oscillator that powers the chip before the crystal oscillator come up during the boot. This RC Oscillator is also the backup Oscillator when the crystal oscillator fails.

The Clocks generated by the PLLs & Oscillators are routed to various entities of the SOC by the use of clock multiplexors. To provide operating frequency of the clocks to various entities of the chip the clocks are divided with the help of the clock dividers.

The clock configuration involves configuring the PLLs, the multiplexors and the dividers at the right time in the boot sequence of the chip. Typically, the clock configuration is handled in the SBL (Secondary Boot loader) and configurations are retained through the lifecycle of the application. Though re-configurations of the clocks can be done at run time keeping in mind of the limitations of each PLL, it's not recommended to re-configure the clocks at run time.



### 5.3.2 Root Clocks

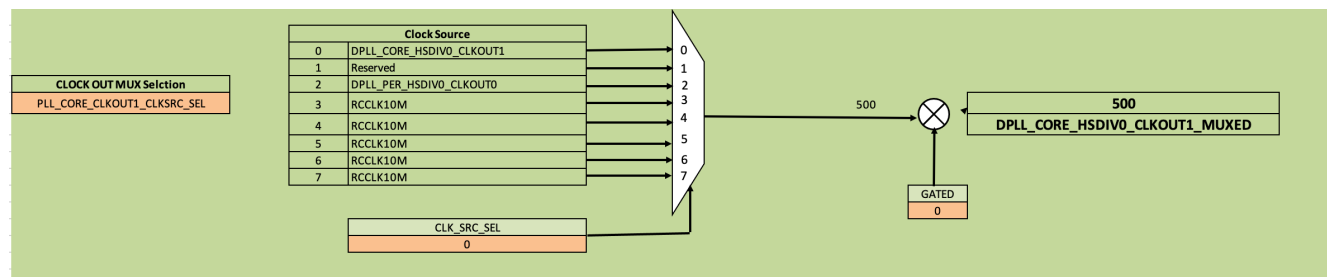
These are the clocks used as inputs to various clock multiplexers and eventually routed to peripherals

**Table 5-1479.**

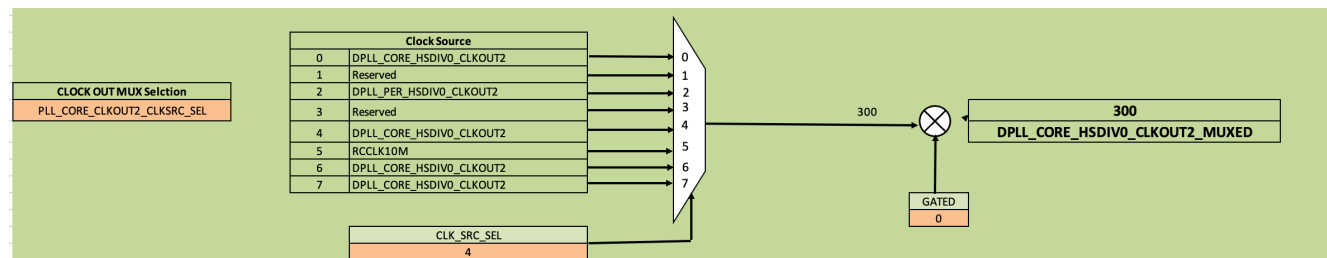
Clock Name	Max Frequency (MHz)	Note
DPLL_CORE_HSDIV0_CLKOUT0	0	
DPLL_CORE_HSDIV0_CLKOUT1	500	
DPLL_CORE_HSDIV0_CLKOUT2	300	
DPLL_CORE_HSDIV0_CLKOUT3	0	
DPLL_PER_HSDIV0_CLKOUT0	0	
DPLL_PER_HSDIV0_CLKOUT1	200	
DPLL_PER_HSDIV0_CLKOUT2	0	
DPLL_PER_HSDIV0_CLKOUT3	0	
RCCLK32K	0.032	Internal fixed clock
RCCLK10M	10	Internal fixed clock
XTALCLK	50	From Crystal
MSS_CR5F_CLK	300	Generated Clock
SYS_CLK	150	Generated Clock
DPLL_CORE_HSDIV0_CLKOUT1_MUXED	500	Generated Clock
DPLL_CORE_HSDIV0_CLKOUT2_MUXED	300	Generated Clock
DPLL_PER_HSDIV0_CLKOUT1_MUXED	200	Generated Clock
WUCPUCLK	50	From XTAL
PLL_CORE_CLK	1500	
PLL_PER_CLK	1000	

### Muxed Primary Clocks

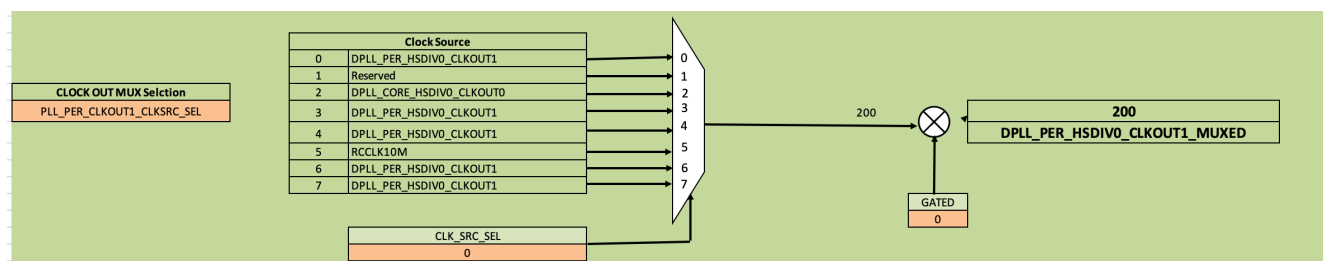
There are 3 primary clocks that are muxed with other clocks. The mux inputs and clock gates for these clocks are as shown in the figures below.



**Figure 5-3. DPLL\_CORE\_HSDIV0\_CLKOUT1\_MUXED**



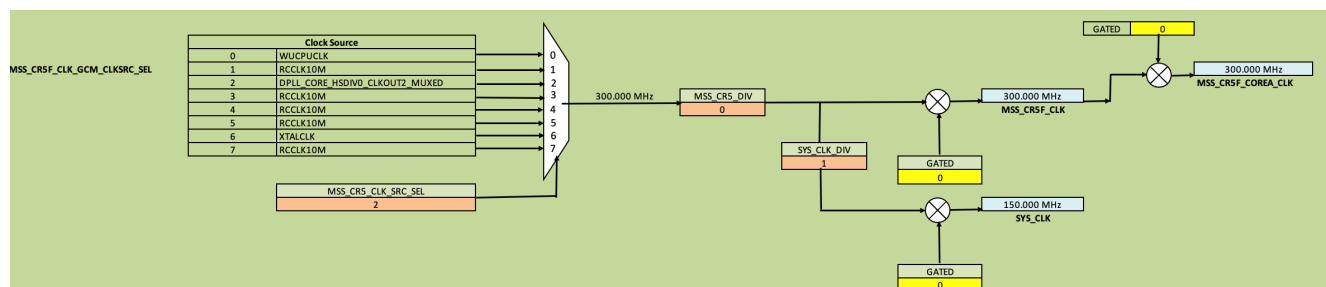
**Figure 5-4. DPLL\_CORE\_HSDIV0\_CLKOUT2\_MUXED**


**Figure 5-5. DPLL\_PER\_HSDIV0\_CLKOUT1\_MUXED**

The registers that control these muxes are as below

**Table 5-1480.**

Name	Address	Bit Position	Typical Value
PLLC_CLK2_SRC_SEL	0x02140464	11:0	
PLLD_CLK1_SRC_SEL	0x02140468	11:0	
PLL_P_CLK1_SRC_SEL	0x02140470	11:0	
PLLC_CLK2_GATE	0x0214047C	2:0	0x0
PLLD_CLK1_GATE	0x02140480	2:0	0x0
PLL_P_CLK1_GATE	0x02140488	2:0	0x0


**Figure 5-6. CR5 and Sys Clock Mux**

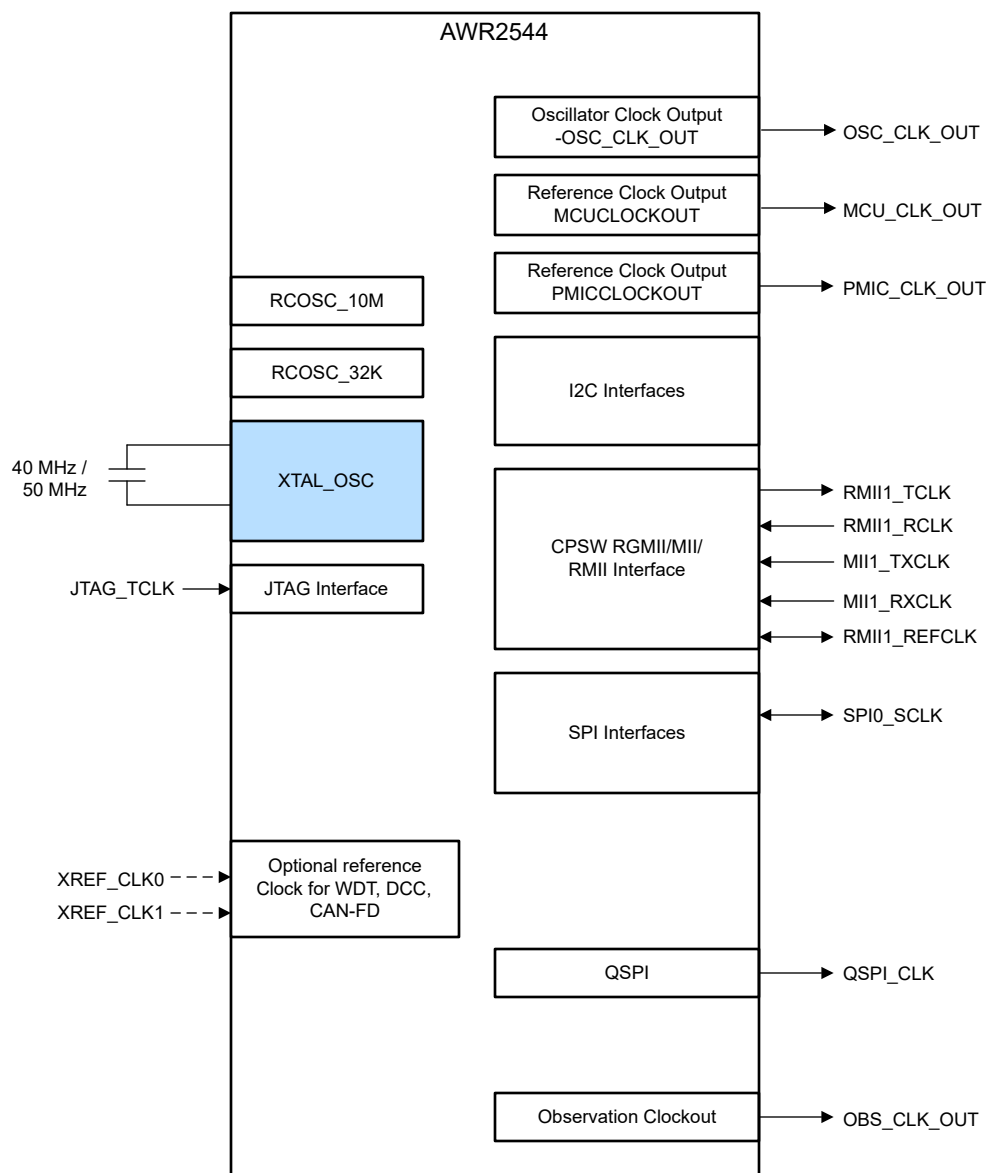
Registers to configure this clock

**Table 5-1481.**

Name	Address	Bit Position	Typical Value
MSS_CR5_CLK_SRC_SEL	0x0214 043C	11:0	0
MSS_CR5_DIV_VAL	0x0214 0440	11:0	0x0
SYS_CLK_DIV_VAL	0x0214 0444	11:0	0x111
MSS_CR5_CLK_GATE	0x0214 0448	2:0	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx
SYS_CLK_GATE	0x0214 044C	2:0	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx
R5_COREA_GATE	0x0210 01B4	2:0	0bxxxx xxxx xxxx xxxx xxxx xxxx xxxx

### 5.3.3 External Clocks

The clocks in the AWR2544 device to the supported peripherals are as depicted in [Figure 5-7](#).



**Figure 5-7. External Clocks**

**Note**

RCOSC\_10M , RCOSC\_32K : these two clocks are internal and not available for application.

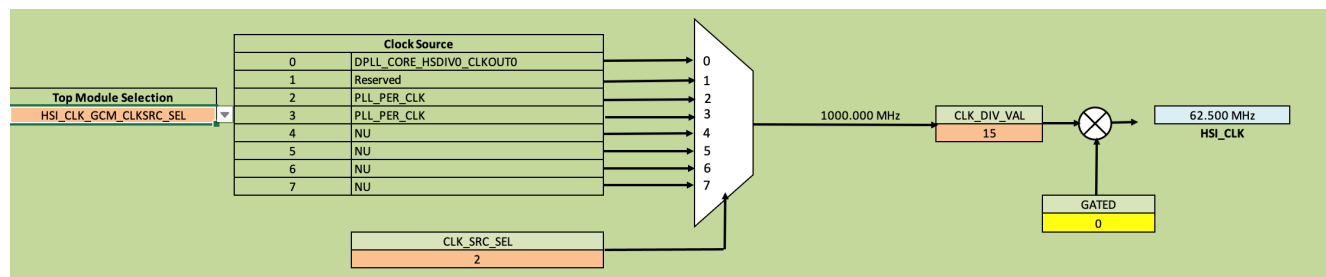
### 5.3.4 Clock Selection

Table 5-1482 lists the configuration options for the clock source, divider, and gating selections for different peripheral clocks.

General structure of the Mux is shown in the figure below.

The general way to configure the clocks are:

1. Go to the clock selection Mux register
2. Select the source for that peripheral, refer the clock source table for the valid sources for that peripheral
3. Program the divider
4. Gate or un gate that clock


**Table 5-1482. Configuration Options**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
MSS_CR5F_COR_EA_CLK	MSS_CR5F_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.1.59</a> [MSS_CR5_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.1.60</a> [MSS_CR5_DIV_VAL_CLKDIV]	<a href="#">Section 5.2.1.62</a> [MSS_CR5_CLK_GATE_GATED]
		1	RESERVED			
		2	DPLL_CORE_HSDIV0_CLKOUT2_MUXED			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	XTALCLK			
		7	RESERVED			
RSS_CLK	RSS_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.1.68</a> [RSS_CLK_SRC_SEL]	<a href="#">Section 5.2.1.73</a> [RSS_DIV_VAL]	<a href="#">Section 5.2.1.74</a> [RSS_CLK_GATE]
		1	XTALCLK			
		2	DPLL_CORE_HSDIV0_CLKOUT2_MUXED			
		3	DPLL_PER_HSDIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	SYS_CLK			
		7	DPLL_CORE_HSDIV0_CLKOUT3			
DPLL_CORE_HSDIV0_CLKOUT2_MUXED	PLL_CORE_CLKOUT2_CLKSRC_SEL	0	DPLL_CORE_HSDIV0_CLKOUT2	<a href="#">Section 5.2.1.69</a> [PLL_CLK2_SRC_SEL_CLKSRCSEL]	NA	<a href="#">Section 5.2.1.75</a> [PLL_CLK2_GATE_GATED]
		1	APLL_1P2_DIV2_CLKOUT			
		2	DPLL_PER_HSDIV0_CLKOUT2			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	RESERVED			
		7	RESERVED			

**Table 5-1482. Configuration Options (continued)**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
DPLL_CORE_HS_DIV0_CLKOUT1_MUXED	PLL_CORE_CLKOUT1_CLKSRC_SEL	0	DPLL_CORE_HSDIV0_CLKOUT1	<a href="#">Section 5.2.1.69</a> [PLLC_CLK2_SRC_SEL_CLKSRCSEL]	NA	<a href="#">Section 5.2.1.75</a> [PLLC_CLK2_CLK_GATE_GATED]
		1	APLL_1P2_DIV2_CLKOUT			
		2	DPLL_PER_HSDIC0_CLKOUT0			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	RESERVED			
		7	RESERVED			
DPLL_PER_HSDIV0_CLKOUT1_MUXED	PLL_PER_CLKOUT1_CLKSRC_SEL	0	DPLL_PER_HSDIV0_CLKOUT1	<a href="#">Section 5.2.1.69</a> [PLLC_CLK2_SRC_SEL_CLKSRCSEL]	NA	<a href="#">Section 5.2.1.75</a> [PLLC_CLK2_CLK_GATE_GATED]
		1	APLL_1P2_DIV2_CLKOUT			
		2	DPLL_CORE_HSDIV0_CLKOUT0			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	RESERVED			
		7	RESERVED			
HSI_CLK	HSI_CLK_GCM_CLKSRC_SEL	0	DPLL_CORE_HSDIV0_CLKOUT0	<a href="#">Section 5.2.1.6</a> [HSI_CLK_SRC_SEL]	NA	NA
		1	ANA_HSI_CLKTO_LVDS			
		2	PLL_PER_CLK			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	RESERVED			
		7	RESERVED			

**Table 5-1482. Configuration Options (continued)**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
TRC_CLK	TRACE_CLKOUT_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.1.21</a> [TRCCLKOUT_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.1.16</a> [TRCCLKOUT_DIV_VAL_CLKDIV]	<a href="#">Section 5.2.1.21</a> [TRCCLKOUT_CLK_GATE_GATED]
		1	DPLL_CORE_HSDIV0_CLKOUT1			
		2	DPLL_CORE_HSDIV0_CLKOUT2_MUXED			
		3	RESERVED			
		4	DPLL_PER_HSDIV0_CLKOUT3			
		5	RESERVED			
		6	XTALCLK			
		7	RESERVED			
MCU_CLKOUT	MCU_CLKOUT_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.1.8</a> [MCUCLKOUT_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.1.13</a> [MCUCLKOUT_DIV_VAL_CLKDIV]	<a href="#">Section 5.2.1.18</a> [MCUCLKOUT_CLK_GATE_GATED]
		1	RESERVED			
		2	DPLL_CORE_HSDIV0_CLKOUT2_MUXED			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	XTALCLK			
		7	RESERVED			
PMIC_CLKOUT	PMIC_CLKOUT_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.1.9</a> [PMICCLKOUT_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.1.14</a> [PMICCLKOUT_DIV_VAL_CLKDIV]	<a href="#">Section 5.2.1.19</a> [PMICCLKOUT_CLK_GATE_GATED]
		1	RESERVED			
		2	DPLL_CORE_HSDIV0_CLKOUT2_MUXED			
		3	RESERVED			
		4	RESERVED			
		5	RESERVED			
		6	XTALCLK			
		7	RESERVED			
OBS_CLKOUT	OBS_CLKOUT_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.1.10</a> [OBSCLKOUT_CLK_SRC_SEL]	<a href="#">Section 5.2.1.15</a> [OBSCLKOUT_DIV_VAL]	<a href="#">Section 5.2.1.20</a> [OBSCLKOUT_CLK_GATE]
		1	DPLL_CORE_HSDIV0_CLKOUT2_MUXED			
		2	DPLL_DSP_HSDIV0_CLKOUT1_MUXED			
		3	DPLL_PER_HSDIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XTALCLK			
		7	RESERVED			

**Table 5-1482. Configuration Options (continued)**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
MSS_RTIA_CLK	MSS_RTIA_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.9</a> [MSS_RTIA_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.20</a> [MSS_RTIA_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.35</a> [MSS_RTIA_CLK_GATE_GATED]
		1	FE1_REF_CLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_RTIB_CLK	MSS_RTIB_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.10</a> [MSS_RTIB_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.21</a> [MSS_RTIB_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.36</a> [MSS_RTIB_CLK_GATE_GATED]
		1	FE1_REF_CLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_RTIC_CLK	MSS_RTIC_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.11</a> [MSS_RTIC_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.22</a> [MSS_RTIC_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.37</a> [MSS_RTIC_CLK_GATE_GATED]
		1	FE1_REF_CLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_WDT_CLK	MSS_WDT_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.12</a> [MSS_WDT_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.23</a> [MSS_WDT_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.38</a> [MSS_WDT_CLK_GATE_GATED]
		1	FE1_REF_CLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			

**Table 5-1482. Configuration Options (continued)**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
MSS_QSPI_CLK	MSS_QSPI_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.8</a> [MSS_QSPI_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.19</a> [MSS_QSPI_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.34</a> [MSS_QSPI_CLK_GATE_GATED]
		1	RESERVED			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	DPLL_CORE_H SDIV0_CLKOUT2_MUXED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_SPIB_CLK	MSS_SPIB_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.13</a> [MSS_SPIB_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.24</a> [MSS_SPIB_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.39</a> [MSS_SPIB_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	DPLL_CORE_H SDIV0_CLKOUT2_MUXED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_I2C_CLK	MSS_I2C_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.14</a> [MSS_I2C_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.25</a> [MSS_I2C_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.40</a> [MSS_I2C_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	DPLL_CORE_H SDIV0_CLKOUT2_MUXED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_SCIA_CLK	MSS_SCIA_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.15</a> [MSS_SCIA_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.26</a> [MSS_SCIA_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.41</a> [MSS_SCIA_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	DPLL_CORE_H SDIV0_CLKOUT2_MUXED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			



**Table 5-1482. Configuration Options (continued)**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
MSS_SCIB_CLK	MSS_SCIB_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.16</a> [MSS_SCIB_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.27</a> [MSS_SCIB_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.42</a> [MSS_SCIB_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	DPLL_CORE_H SDIV0_CLKOUT2_MUXED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
MSS_CPTS_CLK	MSS_CPTS_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.17</a> [MSS_CPTS_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.28</a> [MSS_CPTS_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.43</a> [MSS_CPTS_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_CORE_H SDIV0_CLKOUT1			
		4	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		5	RESERVED			
		6	RESERVED			
		7	RESERVED			
MSS_CPSW_CLK	MSS_CPSW_CLK_GCM_CLK_SRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.18</a> [MSS_CPSW_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.29</a> [MSS_CPSW_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.44</a> [MSS_CPSW_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	DPLL_CORE_H SDIV0_CLKOUT2_MUXED			
		5	RESERVED			
		6	RESERVED			
		7	RESERVED			
HSM_RTI_CLK	HSM_RTI_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.111</a> [HSM_RTIA_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.116</a> [HSM_RTI_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.121</a> [HSM_RTI_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HS DIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			

**Table 5-1482. Configuration Options (continued)**

Clock Name	Clock Mux	Clock Sources		CLKSRCSEL MMR Control	CLKDIV MMR Control	CLKGATE.MMR Control
HSM_WDT_CLK	HSM_WDT_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.112</a> [HSM_WDT_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.117</a> [HSM_WDT_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.122</a> [HSM_WDT_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HSDIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
HSM_RTC_CLK	HSM_RTC_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.113</a> [HSM_RTC_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.118</a> [HSM_RTC_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.123</a> [HSM_RTC_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HSDIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
HSM_DMTA_CLK	HSM_DMTA_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.114</a> [HSM_DMTA_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.119</a> [HSM_DMTA_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.124</a> [HSM_DMTA_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HSDIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			
HSM_DMTB_CLK	HSM_DMTB_CLK_GCM_CLKSRC_SEL	0	RESERVED	<a href="#">Section 5.2.2.120</a> [HSM_DMTB_CLK_SRC_SEL_CLKSRCSEL]	<a href="#">Section 5.2.2.120</a> [HSM_DMTB_CLK_DIV_VAL_CLKDIVR]	<a href="#">Section 5.2.2.125</a> [HSM_DMTB_CLK_GATE_GATED]
		1	XTALCLK			
		2	SYS_CLK			
		3	DPLL_PER_HSDIV0_CLKOUT1_MUXED			
		4	RESERVED			
		5	RESERVED			
		6	XREF_CLK0			
		7	RESERVED			

### 5.3.5 Analog Modules

#### 5.3.5.1 ADPLL

General features of the ADPLL-J module are:

- Low Jitter Phase-Locked Loop
- RHEA, OCP, and Direct Access interface
- Programmable 8-bit input divider: N
- Programmable 12-bit integer, 18-bit fractional loop multiplier: M
- Programmable 7-bit post divider: M2
- Digital control and loop filter
- User-selectable multiple in-built oscillators for power-jitter optimization
- Primary output clock on digital core domain:  $CLKOUT = (M / (N+1)) * CLKINP * (1/M2)$
- Additional output clock on internal LDO domain:  $CLKOUTLDO = (M / (N+1)) * CLKINP * (1/M2)$
- Internal oscillator clock on internal LDO domain:  $CLKDCOLDO = (M / (N+1)) * CLKINP$
- Output clock gating control: CLKOUTEN / CLKOUTLDOEN / CLKOUTDCOLDOEN
- Digital LOCK indicators for frequency and phase lock
- Fast re-lock
- Input to output bypass on CLKOUT
- Bypass programmable 4-bit divider: N2:  $CLKOUT = CLKINP / (N2 + 1)$
- Optional low frequency bypass clock control: ULOWCLKEN CLKOUT = CLKINPULOW
- Power management modes:
  - Power down
  - Idle bypass
  - Stop Clock-input Bypass
  - Retention
- Output clock Spread spectrum clocking supported

The ADPLLJ is a low jitter PLL with a 2-GHz maximum output. ADPLLJ has a predivide feature which allows user to divide, for instance, a 24-MHz or 26-MHz reference clock to 1 MHz and then multiply up to 2 GHz maximum. All PLLs will come-up in bypass mode at reset. SW needs to program all the PLL settings appropriately and then wait for PLL to be locked.

## Spread Spectrum Clocking

The module supports spread spectrum clocking (SSC) on its output clocks. SSC is used to spread the spectral peaking of the clock to reduce any electromagnetic interference (EMI) that may be caused due to the clock's fundamental or any of its harmonics. When SSC is enabled the clock's spectrum is spread by the amount of frequency spread, and the attenuation is given by the ratio of the frequency spread ( $\Delta f$ ) and the modulation frequency ( $f_m$ ), i.e.,  $10 * \log_{10}(\Delta f / f_m)$  dB.

### Note

Please refer [MMWAVE-MCUPLUS-SDK](#) for reference implementation of this feature.

### 5.3.5.2 ADPLLJ Programming Model

#### 5.3.5.2.1 ADPLLJ Programming Sequence

The following sequence is for PLL\_PER. Similar registers exist for PLL\_CORE .

1. Program the values M and N2 in TOP\_RCM: PLL\_PER\_MN2DIV.
2. Program the values of N and M2 in TOP\_RCM:PLL\_PER\_M2NDIV.
3. Program the Fractional M in TOP\_RCM:PLL\_PER\_FRACDIV\_FRACTIONALM.
4. Program the Sigma Delta Divider (SD) in TOP\_RCM:PLL\_PER\_FRACDIV\_REGSD.
5. Copy the NWEELL TRIM from EFUSE to TOP\_RCM:PLL\_PER\_CLKCTRL\_NWEELLTRIM.
6. Write 0x0 to TOP\_RCM:PLL\_PER\_CLKCTRL\_IDLE.
7. Write 0x1 to TOP\_RCM:PLL\_PER\_TENABLE.
8. Write 0x1 to TOP\_RCM:PLL\_PER\_CLKCTRL\_TINTZ.
9. Write 0x0 to TOP\_RCM:PLL\_PER\_TENABLE.
10. Write 0x1 to TOP\_RCM: PLL\_PER\_TENABLEDIV.
11. Write 0x0 to TOP\_RCM: PLL\_PER\_TENABLEDIV.

12. Poll Lock Status by reading TOP\_RCM:PLL\_PER\_STATUS\_PHASELOCK until it is 0x1.

## 5.4 HS Divider Programming Sequence

The following sequence is for PLL\_PER. Similar registers exist for PLL\_CORE .

1. Program the CLKOUT0 Divider value in TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT0\_DIV.
2. Program the CLKOUT0 Divider value in TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT1\_DIV.
3. Program the CLKOUT0 Divider value in TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT2\_DIV.
4. Program the CLKOUT0 Divider value in TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT3\_DIV.
5. Write 0x1 to TOP\_RCM:PLL\_PER\_HSDIVIDER\_TENABLEDIV.
6. Write 0x0 to TOP\_RCM:PLL\_PER\_HSDIVIDER\_TENABLEDIV.
7. Enable the CLKOUT0 by write 0x1 to TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT0\_GATE\_CTRL.
8. Enable the CLKOUT0 by write 0x1 to TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT1\_GATE\_CTRL.
9. Enable the CLKOUT0 by write 0x1 to TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT2\_GATE\_CTRL.
10. Enable the CLKOUT0 by write 0x1 to TOP\_RCM:PLL\_PER\_HSDIVIDER\_CLKOUT3\_GATE\_CTRL.

## 5.5 M2 and N2 Change On-the-Fly

The dividers M2 and N2 are designed to change on the fly and provide a glitch-free frequency switch from the old to new frequencies. In other words, they can be changed while the PLL is in a locked condition, without having to switch to bypass mode. A status toggle bit will give an indication if the new divisor was accepted. These dividers can also be changed in bypass mode, and the new divisor value will be reflected on output after the PLL relocks. For more details, see the PLL configuration procedures for each PLL.

## 5.6 Clock Register Details

**Table 5-1483. Registers for the COREPLL**

Name	Address	Bit Position	Typical Value
PLL_CORE_M2NDIV.N	0x0214 0410	7:0	0
PLL_CORE_M2NDIV.M2	0x0214 0410	22:16	1
PLL_CORE_MN2DIV.N2	0x0214 0414	19:16	0
PLL_CORE_MN2DIV.M	0x0214 0414	11:0	0
PLL_CORE_FRACDIV_REGSD	0x0214 0418	31:24	6
PLL_CORE_FRACDIV_FRACTI ONALM	0x0214 0418	17:0	0
PLL_CORE_HSDIVIDER_CLKO UT0.DIV	0x0214 042C	4:0	Not used
PLL_CORE_HSDIVIDER_CLKO UT1.DIV	0x0214 0430	4:0	Not used
PLL_CORE_HSDIVIDER_CLKO UT2.DIV	0x0214 0434	4:0	Not used
PLL_CORE_HSDIVIDER_CLKO UT3.DIV	0x0214 0438	4:0	Not used
PLL_CORE_HSDIVIDER_CLKO UT0.GATE_CTRL	0x0214 042C	8	1
PLL_CORE_HSDIVIDER_CLKO UT1.GATE_CTRL	0x0214 0430	8	1
PLL_CORE_HSDIVIDER_CLKO UT2.GATE_CTRL	0x0214 0434	8	1
PLL_CORE_HSDIVIDER_CLKO UT3.GATE_CTRL	0x0214 0438	8	1

**Table 5-1484. Registers for the PERPLL**

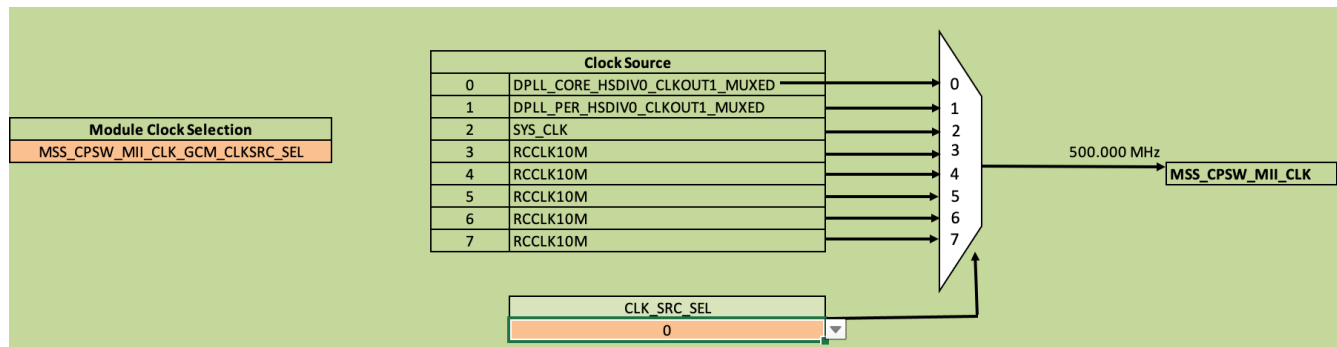
Name	Address	Bit Position	Typical Value
PLL_PER_M2NDIV.N	0x0214 084C	7:0	0

**Table 5-1484. Registers for the PERPLL (continued)**

Name	Address	Bit Position	Typical Value
PLL_PER_M2NDIV.M2	0x0214 084C	22:16	1
PLL_PER_MN2DIV.N2	0x0214 0850	19:16	0
PLL_PER_MN2DIV.M	0x0214 0850	11:0	0
PLL_PER_FRACDIV_REGSD	0x0214 0854	31:24	0
PLL_PER_FRACDIV_FRACTION ALM	0x0214 0854	17:0	4
PLL_PER_HSDIVIDER_CLKOUT 0.DIV	0x0214 0868	4:0	Not used
PLL_PER_HSDIVIDER_CLKOUT 1.DIV	0x0214 086C	4:0	Not used
PLL_PER_HSDIVIDER_CLKOUT 2.DIV	0x0214 0870	4:0	Not used
PLL_PER_HSDIVIDER_CLKOUT 3.DIV	0x0214 0874	4:0	Not used
PLL_PER_HSDIVIDER_CLKOUT 0.GATE_CTRL	0x0214 0868	8	1
PLL_PER_HSDIVIDER_CLKOUT 1.GATE_CTRL	0x0214 086C	8	1
PLL_PER_HSDIVIDER_CLKOUT 2.GATE_CTRL	0x0214 0870	8	1
PLL_PER_HSDIVIDER_CLKOUT 3.GATE_CTRL	0x0214 0874	8	1

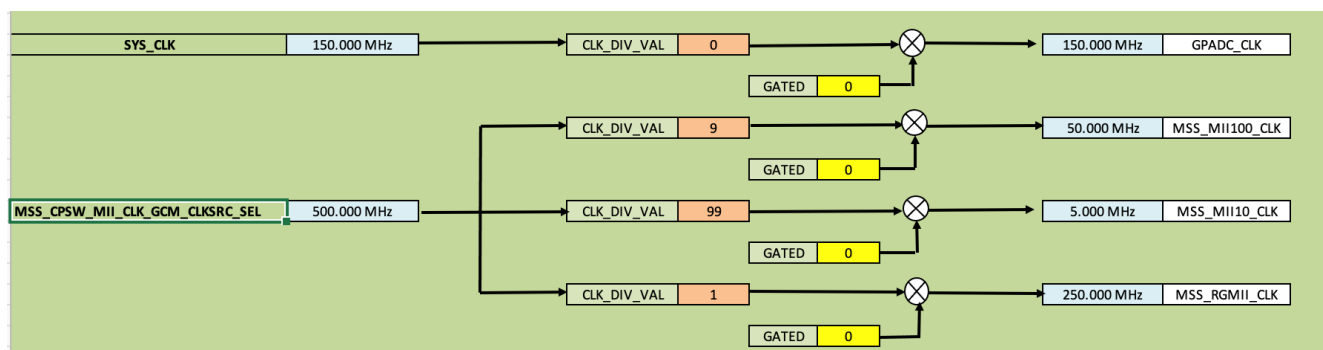
## 5.7 Ethernet Clock

The mux for the ethernet clock configuration is as below


**Table 5-1485. Register for Configuration**

Name	Address	Bit Position	Typical Value
MSS_CPSW_MII_CLK_SRC_SEL	0x021001E0	11:0	0x000

## Ethernet Other Clocks

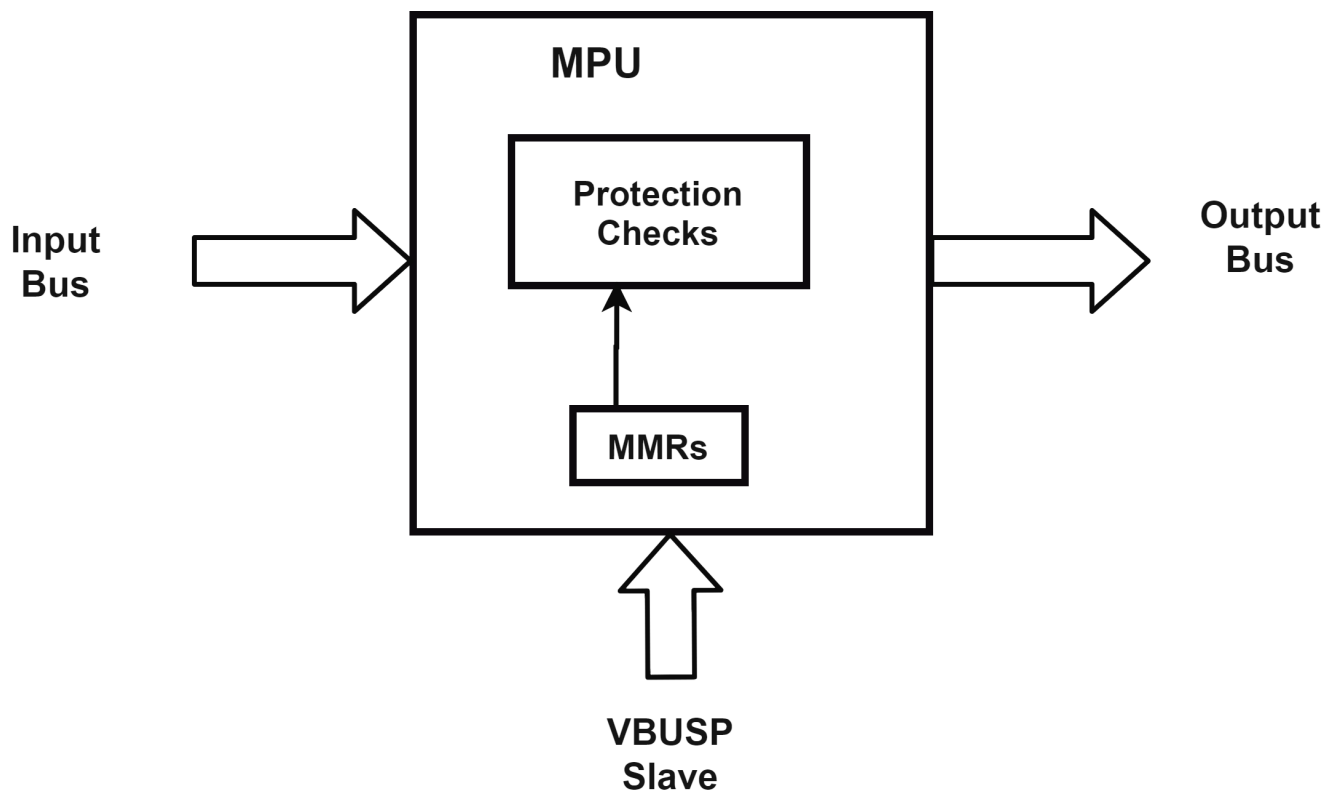

**Table 5-1486. Registers for Configuration**

Name	Address	Bit Position	Typical Value
MSS_RGMII_CLK_DIV_VAL	0x0210 008C	11:0	0x333
MSS_MII100_CLK_DIV_VAL	0x0210 0090	11:0	0x999
MSS_MII10_CLK_DIV_VAL	0x0210 0094	23:0	0x636363
MSS_GPADC_CLK_DIV_VAL	0x0210 0098	23:0	0x0
MSS_RGMII_CLK_GATE	0x0210 00D4	2:0	0bxxxx xxxx xxxx xxxx xxx xxxx x000
MSS_MII100_CLK_GATE	0x0210 00D8	2:0	0bxxxx xxxx xxxx xxxx xxx xxxx x000
MSS_MII10_CLK_GATE	0x0210 00DC	2:0	0bxxxx xxxx xxxx xxxx xxx xxxx x000
MSS_GPADC_CLK_GATE	0x0210 00E0	2:0	0bxxxx xxxx xxxx xxxx xxx xxxx x000

## 5.8 Memory Protection Unit (MPU) and Firewalls

### 5.8.1 Functional Operation

The MPU performs memory protection checking for a CBA bus. It inputs a VBUSM or VBUSP bus, then checks the address against the fixed and programmable regions to see if the access is allowed. If allowed, the transfer is passed unmodified to the output VBUSM or VBUSP bus. If the transfer is illegal (fails the protection check), then the MPU does not pass the transfer to the output bus but rather services the transfer internally back to the input bus (to prevent a hang) returning the fault status to the requestor as well as generating an interrupt about the fault.



**Figure 5-8. MPU Top Level Diagram**

The first check is the transfer's privID against the AID settings. There is an AID register bit for each possible privID (0 to 15) and an AIDX that covers privIDs not configured. The privID is used to lookup the associated AID bit. If the AID bit is 0, then the range does not cover that privilege level and the range is not checked (although other ranges with different AID setting will) for this transfer. If the AID bit is 1, then the range does cover that privilege level and the permissions are checked. The transfer secure and debug parameters are checked against the MPPA values to detect an allowed access. The two bits (NS and EMU) provide 3 permission levels. If the NS is set, the range is non-secure and any security or debug level may access the range. If the NS is not set, the range is secure only and only secure level accesses are allowed. In secure mode, if the EMU is set, debug accesses are allowed. If the EMU is not set, debug accesses are not allowed. For non-debug accesses, the read, write, and execute permissions are also checked. There is a set of permissions for supervisor mode and another for user mode. The "priv" attribute of the transfer determines which is checked. If priv = 1, the supervisor rwx bits are checked against the "dir" and "dtype" attributes of the transfer (read is dir = 1 and dtype not instruction, write is dir = 0, execute is dir = 1 and dtype = instruction). If priv = 0, the user rwx bits are checked against the same attributes. If the associated rwx bit for the type of transfer is 1, the transfer is allowed, but if the rwx bit is 0, the transfer is not allowed.

The function outputs whether the transfer is allowed or not. If the transfer address range (start to end address, or those with AID bits = 0) does not match any range, the transfer is either allowed or disallowed based on the configuration mode of the MPU to "assumed allowed" or "assumed disallowed" mode. If any of the overlapped ranges does not allow the access, the access is not allowed. Only when all the overlapped ranges allow the access is the access allowed. The final permissions are the lowest of each type of permission from any hit range (so if a transfer hits 2 ranges, one that is rw and one that is rx, then the final permission is just r).

### 5.8.2 Firewalls

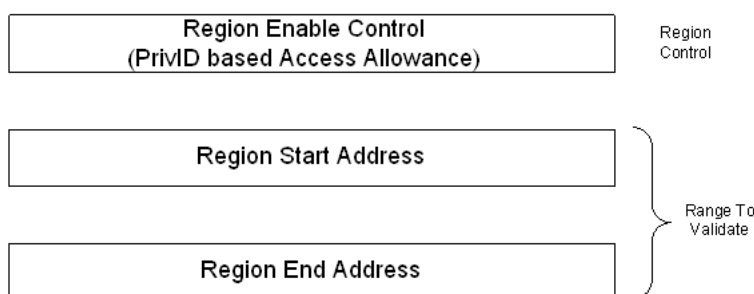
There are 2 types of firewalls in the system: access control firewalls and PrivID-based firewalls. The following sections clarify the usage of the firewalls.

**Access control firewalls** — These firewalls generally allow or disallow access to a particular entity. The different variations of these firewalls are:

- Configuration-based complete (read-write) access allow/disallow
- Configuration-based selected (read/write/read-write) access allow/disallow

The firewalls also support a “Lock” of the configuration, which is a sticky bit implementation that does not allow for any further updates to the configuration until the next power cycle.

**PrivID based firewalls** — The PrivID-based firewall is used to protect peripheral ports from unintentional access from controller ports. The transactions over the bus interconnect have an associated Privilege Identifier field with the read-write access used to determine the validity (allow-disallow) of the transaction. A high level view of the firewall is shown in [Figure 5-9](#).



**Figure 5-9. PrivID Firewall Overview**

A region and the region controls are defined as to which controllers/PrivIDs would be allowed or disallowed access (based on modes). If the address range (start address to end address) of the input transfer overlaps with one of the regions, then the transfer parameters are checked against the address range permissions. These permissions are programmable and follow the MPPA (Memory Protection Permission Attribute) standard set by the Highlander Memory Management architecture. Additionally, the region controls also include read/write/execute permissions both in user/supervisor modes. Further, secure and debug modes can also be configured.

Thus, this section of the PCR controls the configurations of MPU regions in the device. The default\_prog\_mppa, default\_prog\_start\_addr, default\_prog\_end\_addr (for region 0 and 1) at the time of reset. It is controlled based on security manager’s signals soc\_firewall\_bypass and hsm\_firewall\_bypass.

		mppa0	start_addr0	end_addr0	mppa1	start_addr1	end_addr1
		[31:0]	[31:0]	[31:0]	[31:0]	[31:0]	[31:0]
MPU_MSS_L2_BANK_A	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000018BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_MSS_L2_BANK_B	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x0000188F	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_MSS_MB_OX	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x0000188F	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_MSS_PC_R_A	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x00001880	0x00000000	0xFFFFFFFF	0x03FFF400	0x02120800	0x0212FFF



MPU_MSS_QS PI	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x00001880	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_MSS_CR 5A_AXIS	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x0000188F	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_DSS_L3_ BANK_A	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000198BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_DSS_L3_ BANK_B	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000198BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_DSS_L3_ BANK_C	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000198BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_DSS_HW ACC_A	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000198BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_DSS_HW ACC_B	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000198BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_DSS_PR OC (Reserved for future use)	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
	Block	0x000198BF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000
MPU_RSS_MS S2RSS	Firewall Bypass	0x03FFFFFF	0x00000000	0xFFFFFFFF	0x00000000	0x00000000	0x00000000

**Figure 5-10. Default MPU Reset Configurations**

### 5.8.3 ID Allocation

The PrivID allocation is primarily a static allocation in the system, and defined by the values specified in [Table 5-1487](#)

**Table 5-1487. PrivID Grouping and Allocation**

Master Group	PrivID Allocation
MSS Masters	
MSS Cortex R5FA,B AXI-M (x2)	0x2
MSS eDMA TPTCs	0x2
Ethernet DMA	0x2
Debug Masters	
DebugSS JTAG	0x4
RS232	0x4
RCSS Masters	
RCSS ICSS-M Ports (x2)	0x6

**Table 5-1487. PrivID Grouping and Allocation (continued)**

Master Group	PrivID Allocation
RCSS eDMA TPTCs (x2)	0x6

### 5.8.4 Memory Map (MPU)

The default registers are listed in [Table 5-1488](#). There can also be configured MMRs.

**Table 5-1488. Memory Map Registers**

Address Offset	Register
0x000	Revision
0x004	Configuration
0x010	Interrupt Raw Status/Set
0x014	Interrupt Enabled Status/Clear
0x018	Interrupt Enable
0x01C	Interrupt Enable Clear
0x020 - 0x0FC	Reserved
0x100	Reserved
0x104	Reserved
0x108	Reserved
0x10C	Reserved
0x110 - 0x1FC	Reserved
0x200	Programmable 1 Start Address
0x204	Programmable 1 End Address
0x208	Programmable 1 MPPA
0x20C	Reserved
0x210 - 0x2FC	Additional Programmable Range MMRs
0x300	Fault 1 Address
0x304	Fault 1 Status
0x308	Fault 1 Clear

#### 5.8.4.1 Revision Register (Base Address + 0x000)

The Revision Register contains the ID and revision information.

**Table 5-1489. Revision Register**

Bits	Field	Type	Reset	Description
31:30	scheme	r/o	1	Scheme.
29:28	reserved	r/o	0	Always read as 0. Writes have no affect.
27:16	modID	r/o	0xe81	Module ID field.
15:11	revrtl	r/o	Any	RTL revision. Will vary depending on release.
10:8	revmaj	r/o	1	Major revision.
7:6	revcustom	r/o	0	Custom revision.
5:0	revmin	r/o	2	Minor revision.

#### 5.8.4.2 Configuration Register (Base Address + 0x004)

The readonly Configuration register contains the configured values of the module.

**Table 5-1490. Configuration Register**

Bits	Field	Type	Reset	Description
31:24	address_align	r/o	0	Address alignment for range checking. 0 = 1k 1 = 2k 2 = 4k ...
23:20	num_fixed	r/o	0	Number of fixed address ranges.
19:16	num_prog	r/o	8	Number of programmable address ranges.
15:12	num_fixed_aids	r/o	0	Number of supported fixed AIDs. 0 = no specific fixed AIDs supported (all treated equally) N = PrivIDs from 0 to N-1 supported, others use AIDX
11:1	reserved	r/o	0	Always read as 0.
0	assumed_allowed	r/o	0	Assumed allowed mode. 0 = assumed disallowed 1 = assumed allowed

#### 5.8.4.3 Interrupt Raw Status/Set Register (Base Address + 0x010)

The Interrupt Raw Status/Set register shows the interrupt status (before enabling) and allows setting of the interrupt status.

**Table 5-1491. Interrupt Raw Status/Set Register**

Bits	Field	Type	Reset	Description
31:02:00	reserved	r/o	0	Always read as 0.
1	addr_err	w1ts	0	Addressing violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.
0	prot_err	w1ts	0	Protection violation error. Raw status is read. Write a 1 to set the status. Writing a 0 has no effect.

#### 5.8.4.4 Interrupt Enabled Status/Clear Register (Base Address + 0x014)

The Interrupt Enabled Status/Clear register shows the interrupt enabled status and allows clearing of the interrupt status.

**Table 5-1492. Interrupt Enabled Status/Clear Register**

Bits	Field	Type	Reset	Description
31:2	reserved	r/o	0	Always read as 0.
1	enabled_addr_err	w1tc	0	Addressing violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.
0	enabled_prot_err	w1tc	0	Protection violation error. Enabled status is read. Write a 1 to clear the status. Writing a 0 has no effect.

#### 5.8.4.5 Interrupt Enable Register (Base Address + 0x018)

The Interrupt Enable register shows the interrupt enable value and allows setting the enable.

**Table 5-1493. Interrupt Enable Register**

Bits	Field	Type	Reset	Description
31:02	reserved	r/o	0	Always read as 0.

**Table 5-1493. Interrupt Enable Register (continued)**

Bits	Field	Type	Reset	Description
1	addr_err_en	w1ts	0	Addressing violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.
0	prot_err_en	w1ts	0	Protection violation error enable. Write a 1 to set the enable. Writing a 0 has no effect.

**5.8.4.6 Interrupt Enable Clear Register (Base Address + 0x01C)**

The Interrupt Enable Clear register shows the interrupt enable and allows clearing of the interrupt enable. Reads return the Interrupt Enable register value, but writes can clear the interrupt enables.

**Table 5-1494. Interrupt Enable Clear Register**

Bits	Field	Type	Reset	Description
31:02	reserved	r/o	0	Always read as 0.
1	addr_err_en_clr	w1tc	0	Addressing violation error enable. Write a 1 to clear the enable. Writing a 0 has no effect.
0	prot_err_en_clr	w1tc	0	Protection violation error enable. Write a 1 to clear the enable. Writing a 0 has no effect.

**5.8.4.7 EOI Register (Base Address + 0x020)**

The EOI register allows software to indicate when the end of interrupt service is complete. The eoi vector value is dependent on the interrupt handling.

**Table 5-1495. EOI Register**

Bits	Field	Type	Reset	Description
31:8	reserved	r/o	0	Always read as 0.
7:0	eoi_vector	r/w	0	EOI vector value. Write this with the interrupt distribution value in the chip. This drives the mpu_eoi_vector output signal.

**5.8.4.8 Interrupt Vector Register (Base Address + 0x024)**

The Interrupt Vector register displays the interrupt vector returned from interrupt distribution.

**Table 5-1496. Interrupt Vector Register**

Bits	Field	Type	Reset	Description
31:0	intr_vec	r/o	0	Interrupt vector. Reads mpu_intr_vector input signal.

**5.8.4.9 Programmable Address Start Register N (Base Address + 0x200, 0x210, 0x220, ...)**

The Programmable Address Start register holds the start address for the range. This register is writeable by a supervisor entity (priv = 1) only. If the NS bit is in non-secure mode (= 0) in the associated MPPA register, the register is also only writeable by a secure entity (secure = 1). The width of the field is determined by the comp\_width parameter.

(Note: For GP device 'Bypass' is the default reset value and for HS-SE device 'Block' is the default reset value. For MPU\_RSS\_DSS2RSS and MPU\_RSS\_MSS2RSS the default reset value is 'Bypass' for all type of devices)

Refer to Default MPU Reset Configurations.

**Table 5-1497. Programmable Address Start Register**

Bits	Field	Type	Reset	Description
31:N	start_addrN	r/w	input	Start address for range N. Defaults to input signal value.
N-1:0	reserved	r/o	0	Always read as 0.

#### 5.8.4.10 Programmable Address End Register N (Base Address + 0x204, 0x214, 0x224, ...)

The Programmable Address End register holds the end address for the range. This register is writeable by a supervisor entity (priv = 1) only. If the NS bit is in non-secure mode (= 0) in the associated MPPA register, the register is also only writeable by a secure entity (secure = 1). The field width is determined by the comp\_width parameter.

(Note: For GP device 'Bypass' is the default reset value and for HS-SE device 'Block' is the default reset value. For MPU\_RSS\_DSS2RSS and MPU\_RSS\_MSS2RSS the default reset value is 'Bypass' for all type of devices)

Refer to Default MPU Reset Configurations.

**Table 5-1498. Programmable Address End Register**

Bits	Field	Type	Reset	Description
31:N	end_addrN	r/w	input	End address for range N. Defaults to input signal value.
N-1:0	reserved	r/o	1s	Always read as all bits 1.

#### 5.8.4.11 Programmable MPPA Register N (Base Address + 0x208, 0x218, 0x228, ...)

The Programmable Address MPPA register holds the permissions for the region. This register is writeable by a non-debug supervisor entity (priv = 1) only. If the NS bit is in secure mode (= 0), the register is also only writeable by a non-debug secure entity (secure = 1). The NS bit is only writeable by a non-debug secure entity. For debug accesses, the register is writeable only when NS = 1 or EMU = 1.

(Note: For GP device 'Bypass' is the default reset value and for HS-SE device 'Block' is the default reset value. For MPU\_RSS\_DSS2RSS and MPU\_RSS\_MSS2RSS the default reset value is 'Bypass' for all type of devices)

Refer to Default MPU Reset Configurations.

**Table 5-1499. Programmable MPPA Register**

Bits	Field	Type	Reset	Description
31:26	reserved	r/o	0	Always read as 0.
25:10	AID15-0	r/w	input	AIDs checked for this region. Defaults to input value. 0 = AID is not checked for these permissions. 1 = AID is checked for these permissions.
9	AIDX	r/w	input	Additional AIDs checked. Defaults to input value.
8	reserved	r/o	0	Always read as 0.
7	ns	r/w	input	Non-secure permission. Defaults to input value.
6	emu	r/w	input	Debug permission. Defaults to input value.
5	sr	r/w	input	Supervisor read permission. Defaults to input value.
4	sw	r/w	input	Supervisor write permission. Defaults to input value.
3	sx	r/w	input	Supervisor executable permission. Defaults to input value.

**Table 5-1499. Programmable MPPA Register (continued)**

Bits	Field	Type	Reset	Description
2	ur	r/w	input	User read permission. Defaults to input value.
1	uw	r/w	input	User write permission. Defaults to input value.
0	ux	r/w	input	User executable permission. Defaults to input value.

**5.8.4.12 Fault Address Register (Base Address + 0x300)**

The Fault Address register holds the address of the first protection fault transfer.

**Table 5-1500. Fault Address Register**

Bits	Field	Type	Reset	Description
31:0	fault_addr	r/o	0	Fault address.

**5.8.4.13 Fault Status Register (Base Address + 0x304)**

The Fault Status register holds the status and attributes of the first protection fault transfer.

**Table 5-1501. Fault Status Register**

Bits	Field	Type	Reset	Description
31:24	id	r/o	0	Transfer ID.
23:16	mstid	r/o	0	controller ID.
15:13	reserved	r/o	0	Always read as 0.
12:09	privid	r/o	0	Privilege ID.
8	reserved	r/o	0	Always read as 0.
7	ns	r/o	0	Non-secure access.
6	reserved	r/o	0	Always read as 0.
5:00	fault_type	r/o	0	Fault type. 100000 = supervisor read fault 010000 = supervisor write fault 001000 = supervisor execute fault 000100 = user read fault 000010 = user write fault 000001 = user execute fault 111111 = relaxed cache linefill fault 010010 = relaxed cache writeback fault 000000 = no fault

**5.8.4.14 Fault Clear Register (Base Address + 0x308)**

The Fault Clear register allows the software to clear the current fault, so that another can be captured when this register is written.

**Table 5-1502. Fault Clear Register**

Bits	Field	Type	Reset	Description
31:1	reserved	r/o	0	Always read as 0.
0	fault_clr	w/o	0	Fault clear. Writing a 1 clears the current fault. Writing a 0 has no effect.

**5.9 Power Domains**

The DSS\_L3 memory, MSS\_L2 memory and the HWA can be switched off dynamically if required.

### 5.9.1 DSP Subsystem L3 Power Domain

The DSS L3 memory can be controlled from DSS\_RCM registers by the software.

By default, all the L3 memory is Power ON.

When partially switching off banks, it is the software's responsibility to ensure that the banks are disabled from the last 256 KB bank, that is, DSS L3 BANK B1. A bank should not be powered off if all its higher banks are not powered off. For example, if DSS L3 BANK A1 is powered down, BANK B0 and B1 are also powered down.

#### 5.9.1.1 L3 Memory Bank Power Sequence

Below is the sequence to Power off a bank:

1. Write 0x7 to DSS\_RCM:DSS\_L3\_BANK\*\_PD\_CTRL\_ISO field.
2. Write 0x0 to DSS\_RCM:DSS\_L3\_BANK\*\_PD\_CTRL\_AONIN field.
3. Wait until DSS\_RCM:DSS\_L3\_BANK\*\_PD\_STATUS\_AONOUT field is 0x0.
4. Write 0x0 to DSS\_RCM:DSS\_L3\_BANK\*\_PD\_CTRL\_AGOODIN field.
5. Wait until DSS\_RCM:DSS\_L3\_BANK\*\_PD\_STATUS\_AGOODOUT field is 0x0.

Below is the sequence to Power on a bank:

1. Write 0x7 to DSS\_RCM:DSS\_L3\_BANK\*\_PD\_CTRL\_AONIN field.
2. Wait until DSS\_RCM:DSS\_L3\_BANK\*\_PD\_STATUS\_AONOUT field is 0x1.
3. Write 0x7 to DSS\_RCM:DSS\_L3\_BANK\*\_PD\_CTRL\_AGOODIN field.
4. Wait until DSS\_RCM:DSS\_L3\_BANK\*\_PD\_STATUS\_AGOODOUT field is 0x1.
5. Write 0x0 to DSS\_RCM:DSS\_L3\_BANK\*\_PD\_CTRL\_ISO field.

**Table 5-1503. DSS L3 Bank Control**

Bank 256 KB	Bank Name	Control Registers
0	DSS L3 BANK A0	DSS_L3_BANKA0_PD_CTRL_AGOODIN DSS_L3_BANKA0_PD_CTRL_AONIN DSS_L3_BANKA0_PD_CTRL_ISO
1	DSS L3 BANK A1	DSS_L3_BANKA1_PD_CTRL_AGOODIN DSS_L3_BANKA1_PD_CTRL_AONIN DSS_L3_BANKA1_PD_CTRL_ISO
4	DSS L3 BANK B0	DSS_L3_BANKA2_PD_CTRL_AGOODIN DSS_L3_BANKA2_PD_CTRL_AONIN DSS_L3_BANKA2_PD_CTRL_ISO
5	DSS L3 BANK B1	DSS_L3_BANKB0_PD_CTRL_AGOODIN DSS_L3_BANKB0_PD_CTRL_AONIN DSS_L3_BANKB0_PD_CTRL_ISO

### Sticky Bit Implementation

The L3 banks power control can be locked from further updates by writing 0x7 to the register DSS\_RCM:DSS\_L3\_PD\_CTRL\_STICKYBIT:SET field. Any update to the DSS\_L3\_BANK\*\_PD\_CTRL, AONIN and AGOODIN fields do not have an impact on HW. The sticky bit is reset with Warm Reset.

The actual HW control to the memory can be read by reading the fields DSS\_L3\_BANK\*\_PD\_STATUS\_AONIN and DSS\_L3\_BANK\*\_PD\_STATUS\_AGOODIN

### 5.9.2 MSS L2 Power Domain

The MSS L2 memory can be controlled from the MSS\_RCM registers by the software. By default, all the MSS\_L2 memory is Power ON.

### 5.9.2.1 Power Sequence

Below is the sequence to Power off a bank:

1. Write 0x7 to MSS\_RCM:MSS\_L2\_BANK\*\_PD\_CTRL\_ISO field.
2. Write 0x0 to MSS\_RCM:MSS\_L2\_BANK\*\_PD\_CTRL\_AONIN field.
3. Wait until MSS\_RCM:MSS\_L2\_BANK\*\_PD\_STATUS\_AONOUT field is 0x0.
4. Write 0x0 to MSS\_RCM:MSS\_L2\_BANK\*\_PD\_CTRL\_AGOODIN field.
5. Wait until MSS\_RCM:MSS\_L2\_BANK\*\_PD\_STATUS\_AGOODOUT field is 0x0.

Below is the sequence to Power on a bank:

1. Write 0x7 to MSS\_RCM:MSS\_L2\_BANK\*\_PD\_CTRL\_AONIN field.
2. Wait until MSS\_RCM:MSS\_L2\_BANK\*\_PD\_STATUS\_AONOUT field is 0x1.
3. Write 0x7 to MSS\_RCM:MSS\_L2\_BANK\*\_PD\_CTRL\_AGOODIN field.
4. Wait until MSS\_RCM:MSS\_L2\_BANK\*\_PD\_STATUS\_AGOODOUT field is 0x1.
5. Write 0x0 to MSS\_RCM:MSS\_L2\_BANK\*\_PD\_CTRL\_ISO field.

When a block is powered off, a bus error is generated on access. It is always safe to have decent delay between each step because memory might take some time before reaching to total power-on state.

**Table 5-1504. Registers Controls For Each Memory Block**

Bank 512KB	Bank Name	Control Registers
0	MSS L2 BANK A	MSS_L2_BANKA_PD_CTRL_AGOODIN MSS_L2_BANKA_PD_CTRL_AONIN MSS_L2_BANKA_PD_CTRL_ISO
1	MSS L2 BANK B	MSS_L2_BANKB_PD_CTRL_AGOODIN MSS_L2_BANKB_PD_CTRL_AONIN MSS_L2_BANKB_PD_CTRL_ISO

### HWA Power Domain

The HWA can be controlled from DSS\_RCM registers by SW. By Default, all the HWA is Power ON.

#### Power Sequence

Below is the sequence to Power off the HWA

1. Write 0x7 to DSS\_RCM:DSS\_HWA\_PD\_CTRL:ISO field
2. Write 0x0 to DSS\_RCM:DSS\_HWA\_PD\_CTRL:AONIN field
3. Wait till DSS\_RCM:DSS\_HWA\_PD\_STATUS:AONOUT field is 0x0
4. Write 0x0 to DSS\_RCM:DSS\_HWA\_PD\_CTRL:AGOODIN field
5. Wait till DSS\_RCM:DSS\_HWA\_PD\_STATUS:AGOODOUT field is 0x0

Below is the sequence to Power on the HWA

1. Write 0x7 to DSS\_RCM:DSS\_HWA\_PD\_CTRL:AONIN field
2. Wait till DSS\_RCM:DSS\_HWA\_PD\_STATUS:AONOUT field is 0x1
3. Write 0x7 to DSS\_RCM:DSS\_HWA\_PD\_CTRL:AGOODIN field
4. Wait till DSS\_RCM:DSS\_HWA\_PD\_STATUS:AGOODOUT field is 0x1
5. Write 0x0 to DSS\_RCM:DSS\_HWA\_PD\_CTRL:ISO field

It is always safe to have decent delay between each step because HWA might take some time before reaching to total power on state. So even though AONOUT is 1'b1 does not mean HWA is ON.

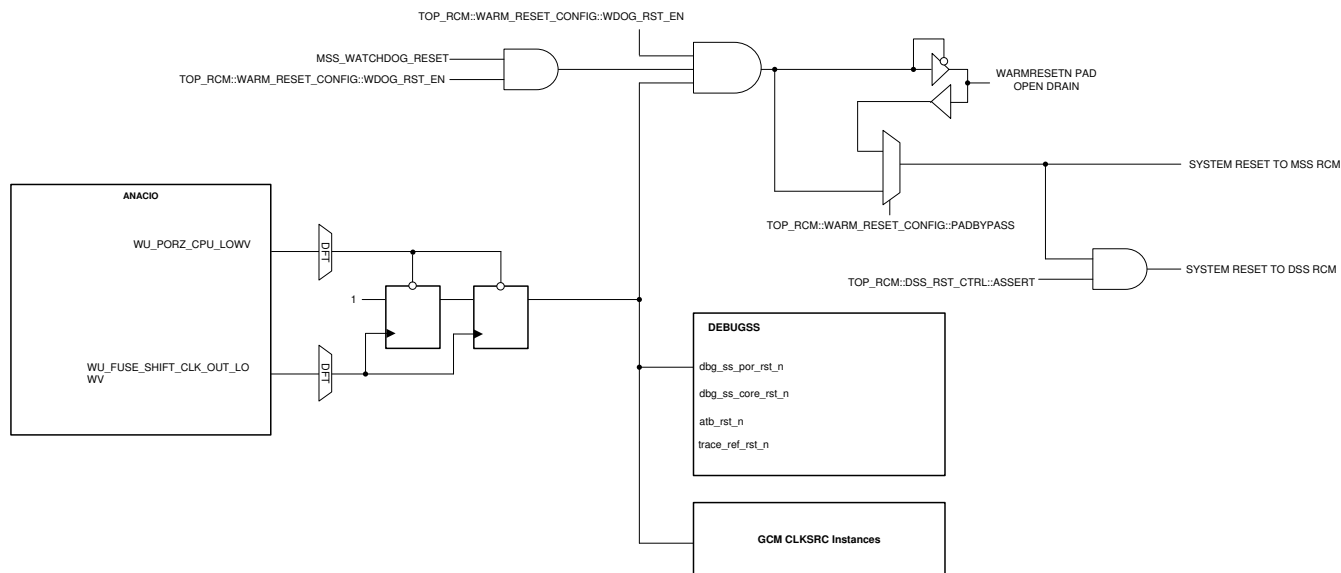
### Registers

Below is the corresponding register controls for each block of memory



Bank 512KB	Bank Name	Control Register
0	HWA (Hard Macro)	DSS_RCM:DSS_HWA_PD_CTRL:AGOODIN DSS_RCM:DSS_HWA_PD_CTRL:AONIN DSS_RCM:DSS_HWA_PD_CTRL:ISO DSS_RCM:DSS_HWA_PD_STATUS

## 5.10 Resets



### Figure 5-11. Resets

Two device resets are available that can be controlled from the device pins: the power-on reset pin NRESET and the bidirectional WARM\_RESET signal. The warm reset signal is implemented as an I/O, so that an external monitor can be used to detect changes to the state of the internal warm reset control signal.

## Reset Types and Sources

### Table 5-1505. Reset Types and Sources

Reset Type	Reset Source	Description
Power On Reset	Device Pin NRESET	Reset triggered by the device reset pin NRESET. This resets the entire device, including all subsystems and interfaces.  This is an active low asynchronous Power ON reset signal, and must be asserted for minimum 20 usec to reset the device.

**Table 5-1505. Reset Types and Sources (continued)**

Reset Type	Reset Source	Description
Warm Reset	Software Reset (Register write), Watch Dog Reset, Device Pin WARM_RESET pin	<p>This is an active low warm reset internally generated by the device, or triggered by device pin WARM_RESET.</p> <p>A write to the TOP_RCM.</p> <p>WARM_RESET_CONFIG[10:8] register or watch dog module can generate this reset; additionally, the external pin WARM_RESET can also be used to trigger this reset. Selection is controlled through the TOP_RCM.WARM_RESET_CONFIG register.</p> <p>The WARM_RESET pin is an open-drain failsafe IO which can be used to reset the device from the external world or to report the reset to the external world if it is generated by an internal source such as watchdog.</p> <p>A write to the TOP_RCM.WARM_RESET_CONFIG, as explained in register description, can create this reset. The watch dog module in the main subsystem can be configured to trigger this reset. Check the watch dog module description for details. This feature is enabled by the TOP_RCM.WARM_RESET_CONFIG register.</p>

As listed in above table, various reset sources can generate the different resets used inside the device to reset various components and submodules. A Warm\_reset excludes certain modules and spaces listed in this table from reset. These can only be reset by power-on reset.

## Reset Domains

The device can be divided into various reset domains. The top reset domains cover the entire device and all of the subsystems. Additional subsystem-level reset domains are available, and can be reset independently based on resets mentioned in below table.

**Table 5-1506. Reset Domains**

Reset Domains	Description	Resets
Top reset domain	This top device-level reset domain resets entire device and all subsystems. All other reset domains are subdomains of this domain, and resetting this domain issues a reset to these subdomains. Only power-on reset can reset this domain, and it is immune to any other system reset type. Sub reset domain can be independently reset, as mentioned in the respective rows.	Power-on reset
Main subsystem reset domain	This reset subdomain controls the reset to the main subsystem and the modules inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset

**Table 5-1506. Reset Domains (continued)**

Reset Domains	Description	Resets
Radar subsystem reset domain	This reset subdomain controls the reset to the radar subsystem and all components inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset
DSP subsystem reset domain	This reset subdomain controls the reset to the DSP subsystem and all components inside it. Resetting the top reset domain also resets this domain.	Power-on reset Warm reset

## 5.11 Temperature Sensor

The AWR2544 device has multiple on-chip temperature sensors (TS) which are distributed in and around different components of silicon. The RF front-end of the device contains multiple TS, which RadarSS can access to measure the temperature for RX and TX components. The application can read these TS through RadarSS using the mmWaveLink API (AWR\_RF\_TEMPERATURE\_GET\_SB, rIRfGetTemperatureReport). There are three digital temperature sensors available on the die nearby specific components [TS0: C66x, TS1: HWA, TS2: R5F] which the application can access via [GPADC](#). There are two digital temperature sensors available on the die nearby specific components [ TS1: HWA, TS2: R5F] which the application can access via [GPADC](#).

### Note

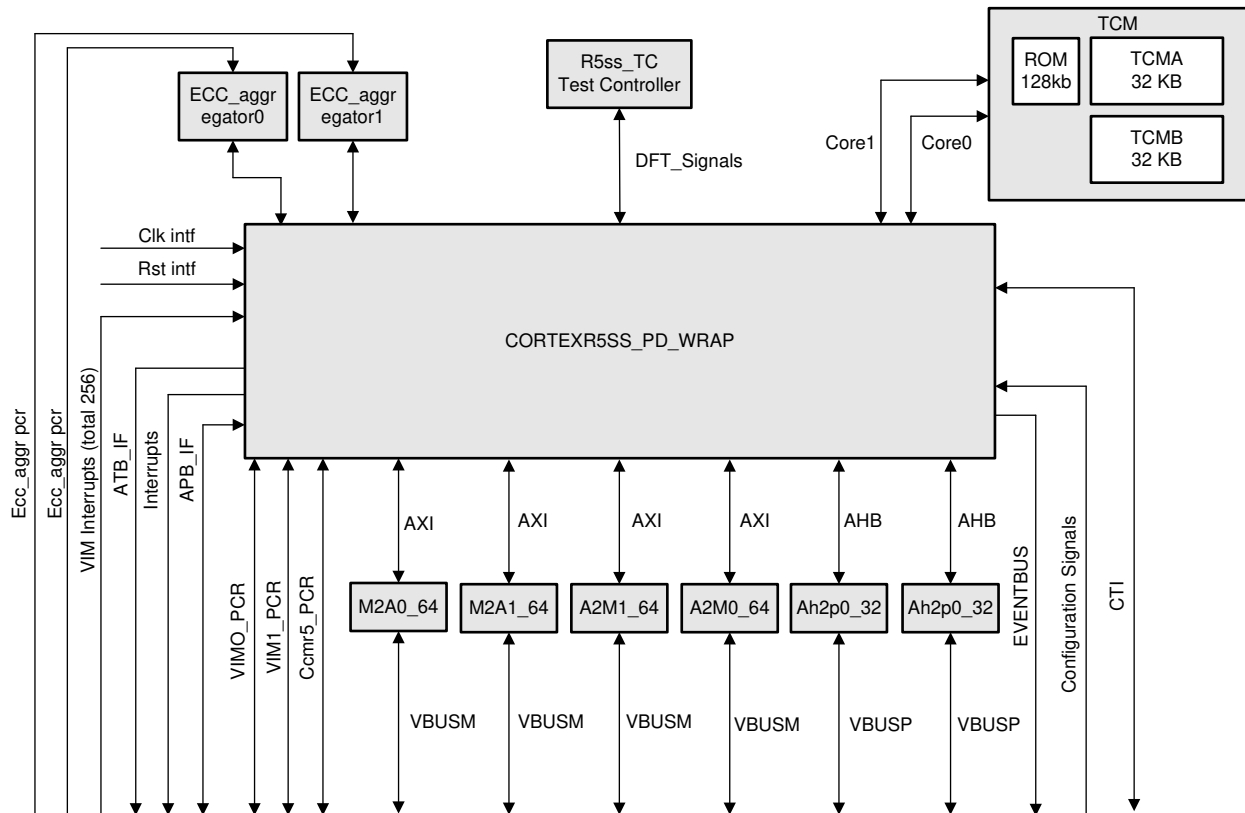
Refer to the GPADC driver and unit-test application from [MMWAVE-MCUPLUS-SDK](#) for reference code to read these temperature sensors.



Device includes the following processor core and accelerators:

- MCU Subsystem ARM Cortex lockstep R5F
- Radar Processing Hardware Accelerators
- Radar Subsystem ARM Corex lockstep R4F

## 6.1 Main Subsystem Cortex R5F



**Figure 6-1. Main Subsystem**

It also includes accompanying memories (L1 caches and tightly-coupled memories), standard Arm® CoreSight™ debug and trace architecture, integrated Vectored Interrupt Manager (VIM), ECC Aggregators, and various other modules for protocol conversion and address translation for easy integration into the SoC.

### 6.1.1 Main ARM SubSystem Features

The MSS supports the following features:

- ARMv7-R architecture with the following extensions:

- Basic SIMD extension for integer and floating-point vector operations
- Vector Floating Point Version 3 (VFPv3) with Single/Double Precision
- Dynamic branch prediction with a global history buffer, and a 4-entry return stack
- Only Lock step mode is supported.
- L1 memory architecture :
  - 16KByte I-Cache with 64-bit ECC
  - 16Kbyte D-cache with 32-bit ECC
  - 128KB TCM with 32-bit ECC
    - in lock-step mode
    - for each core in split-mode
- L2 Interface
  - 64-bit Controller interface for peripheral access
  - 64-bit Target interface for cache and TCM access
  - 32-bit Controller interface for MCU peripherals
  - ECC on data. Redundancy on Control bus
- Vectored Interrupt Manager with ECC protection on Vector Table RAM
  - VIM1 and VIM2 in lockstep pair when CPU cores are in lock step mode
- Static (boot time) configuration for lockstep mode.
- CPU Self-Test Controller for CPU core, VIM, and Comparator modules
- PBIST controller for test of all the RAMs
- Support to test ECC functionality in safety-critical applications
- Built in debug features
  - Up to 8 hardware breakpoints per CPU
  - Up to 8 watch points per CPU
- 32-bit Target Debug interface to access Debug components (CTI, ETM, ATB)
- Trace interface to a Core Sight ETM-R5
- Performance Monitoring Unit (PMU)
- Little Endian mode of operation

### 6.1.2 TCM Initialization

Auto-init module has been implemented for initializing the TCMs. Paths from TCMA and TCMB are timing-critical, so the initialization of these memories occur through the test path. Initialization of TCMA and TCMB occurs in parallel.

Below are the registers used for the TCM initialization

- Writing 1 to MSS\_CTRL:MSS\_<A/B>TCM\_MEM\_INIT:MSS\_<A/B>TCM\_MEM\_INIT\_MEM\_INIT starts the mem-init for MSS\_TCM<A/B>\_CR5A.
- Reading 1 from MSS\_CTRL: MSS\_<A/B>TCM\_MEM\_INIT\_DONE: MSS\_<A/B>TCM\_MEM\_INIT\_DONE\_MEM\_INIT\_DONE confirms the end of initialization for MSS\_TCM<A/B>\_CR5A.
- Writing 1 to MSS\_CTRL: MSS\_<A/B>TCM\_MEM\_INIT\_DONE: MSS\_<A/B>TCM\_MEM\_INIT\_DONE\_MEM\_INIT\_DONE clears the field.
- Reading 1 from MSS\_CTRL: MSS\_<A/B>TCM\_MEM\_INIT\_STATUS: MSS\_<A/B>TCM\_MEM\_INIT\_STATUS\_MEM\_STATUS confirms progress of initialization for MSS\_TCM<A/B>\_CR5A.

### 6.1.3 Resets

AWR2544 supports the following resets:

- CR5SS\_POR\_RST: This is the full MCU R5F subsystem reset. It is also the Power On Reset.
- CR5ASS\_RST: This reset is only for the Cortex R5F and the Vectored Interrupt Manager of the CoreA. None of the other logics are reset.
- CR5A\_RST: This only reset the cortex of the R5F of the CoreA.
- VIMA\_RST: This only reset the vectored interrupt manager of the CoreA.

### 6.1.3.1 R5F Subsystem Reset Trigger

For the safe reset of the R5F subsystem, follow this reset sequence:

1. Write 3'b111 to MSS\_CTRL: R5\_CONTROL: R5\_CONTROL\_RESET\_FSM\_TRIGGER starts the sequencer.
2. The sequencer is waiting for WFI from only CR5A if WFI\_CHECKEN for corresponding resets is programmed. The sequencer isolates CR5SS when it receives WFI from CR5A.
3. The sequencer triggers CR5SS\_POR\_RST, which triggers resets to all resets mentioned in resets-table.
4. The intent is to reset CR5SS when it is in isolation. Thus, before Step 1, ensure MSS\_RCM:RST2ASSERTDLY is programmed to 0;.
5. Reset is asserted for RST\_ASSERTDLY: RST\_ASSERTDLY\_COMMON number of clock cycles, and released.

### 6.1.4 MSS\_CCMR Registers

[Table 6-1](#) lists the memory-mapped registers for the MSS\_CCMR registers. All register offset addresses not listed in [Table 6-1](#) should be considered as reserved locations and the register contents should not be modified.

**Table 6-1. MSS\_CCMR Registers**

Offset	Acronym	Register Name	Section
0h	CCMSR1	CCMSR1	<a href="#">Go</a>
4h	CCMKEYR1	CCMKEYR1	<a href="#">Go</a>
8h	CCMSR2	CCMSR2	<a href="#">Go</a>
Ch	CCMKEYR2	CCMKEYR2	<a href="#">Go</a>
10h	CCMSR3	CCMSR3	<a href="#">Go</a>
14h	CCMKEYR3	CCMKEYR3	<a href="#">Go</a>
18h	CCMPOLCNTRL	CCMPOLCNTRL	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 6-2](#) shows the codes that are used for access types in this section.

**Table 6-2. MSS\_CCMR Access Type Codes**

Access Type	Code	Description
<b>Read Type</b>		
R	R	Read
<b>Write Type</b>		
W	W	Write
<b>Reset or Default Value</b>		
-n		Value after reset or the default value

### 6.1.4.1 CCMSR1 Register (Offset = 0h) [Reset = 00000000h]

CCMSR1 is shown in [Table 6-3](#).

Return to the [Summary Table](#).

CPU Compare Status Register

**Table 6-3. CCMSR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	NU2	R/W	0h	Reserved
16	CMPE1	R/W	0h	Compare Error 0 = CPU signals are identical 1 = CPU signal compare mismatch Writes '1' to clear this bit
15-9	NU1	R/W	0h	Reserved
8	STC1	R/W	0h	Self Test Complete 0 = self test on-going if self test mode asserted 1 = self test is complete Writes have no effect
7-2	NU0	R/W	0h	Reserved
1	STET1	R/W	0h	Self Test Error Type 0 = self test failed during Compare Match test 1 = self test failed during Compare mismatch test Writes have no effect
0	STE1	R	0h	Self Test Error 0 = self test passed 1 = self test failed Writes have no effect



#### 6.1.4.2 CCMKEYR1 Register (Offset = 4h) [Reset = 00000000h]

CCMKEYR1 is shown in [Table 6-4](#).

Return to the [Summary Table](#).

CPU Compare Key Register

**Table 6-4. CCMKEYR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	NU3	R/W	0h	Reserved
3-0	MKEY1	R/W	0h	Mode Key 0000 = lock step mode 0110 = self test mode 1001 = error forcing mode 1111 = self test error forcing mode

### 6.1.4.3 CCMSR2 Register (Offset = 8h) [Reset = 00000000h]

CCMSR2 is shown in [Table 6-5](#).

Return to the [Summary Table](#).

VIM Compare Status Register

**Table 6-5. CCMSR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	NU6	R/W	0h	Reserved
16	CMPE2	R/W	0h	Compare Error 0 = VIM signals are identical 1 = VIM signal compare mismatch Writes '1' to clear this bit
15-9	NU5	R/W	0h	Reserved
8	STC2	R/W	0h	Self Test Complete 0 = self test on-going if self test mode asserted 1 = self test is complete Writes have no effect
7-2	NU4	R/W	0h	Reserved
1	STET2	R/W	0h	Self Test Error Type 0 = self test failed during Compare Match test 1 = self test failed during Compare mismatch test Writes have no effect
0	STE2	R/W	0h	Self Test Error 0 = self test passed 1 = self test failed Writes have no effect

#### 6.1.4.4 CCMKEYR2 Register (Offset = Ch) [Reset = 00000000h]

CCMKEYR2 is shown in [Table 6-6](#).

Return to the [Summary Table](#).

VIM Compare Key Register

**Table 6-6. CCMKEYR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	NU7	R/W	0h	Reserved
3-0	MKEY2	R/W	0h	Mode Key 0000 = lock step mode 0110 = self test mode 1001 = error forcing mode 1111 = self test error forcing mode

#### 6.1.4.5 CCMSR3 Register (Offset = 10h) [Reset = 00000000h]

CCMSR3 is shown in [Table 6-7](#).

Return to the [Summary Table](#).

Inactivity Monitor Status Register

**Table 6-7. CCMSR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	NU10	R/W	0h	Reserved
16	CMPE3	R/W	0h	Compare Error 0 = Inactivity monitor signals are identical 1 = Inactivity monitor signal compare mismatch Writes '1' to clear this bit
15-9	NU9	R/W	0h	Reserved
8	STC3	R/W	0h	Self Test Complete 0 = self test on-going if self test mode asserted 1 = self test is complete Writes have no effect
7-2	NU8	R/W	0h	Reserved
1	STET3	R/W	0h	Self Test Error Type 0 = self test failed during Compare Match test 1 = self test failed during Compare mismatch test Writes have no effect
0	STE3	R	0h	Self Test Error 0 = self test passed 1 = self test failed Writes have no effect

#### 6.1.4.6 CCMKEYR3 Register (Offset = 14h) [Reset = 00000000h]

CCMKEYR3 is shown in [Table 6-8](#).

Return to the [Summary Table](#).

Inactivity Monitor Key Register

**Table 6-8. CCMKEYR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	NU11	R/W	0h	Reserved
3-0	MKEY3	R/W	0h	Mode Key 0000 = lock step mode 0110 = self test mode 1001 = error forcing mode 1111 = self test error forcing mode

### 6.1.4.7 CCMPOLCNTRL Register (Offset = 18h) [Reset = 00000000h]

CCMPOLCNTRL is shown in [Table 6-9](#).

Return to the [Summary Table](#).

CPU Compare Polarity Control Register

**Table 6-9. CCMPOLCNTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU12	R/W	0h	Reserved
7-0	POL_INV	R	0h	This value is used to invert the 8 XOR of the CPU1 to create compare fail in functional active compare mode. User and privilege mode read = Returns current value of the POL INV Privilege mode write = Update the values of POL INV

## 6.2 Radar Processing Hardware Accelerator

The AWR2544 device incorporates Radar Hardware Accelerators HWA1.5 to offload the CPU from pre-processing computations.

### 6.2.1 Radar Accelerator Features Overview

- Flexible enough to offload almost all the operations from the initial compensation (DC estimation-correction, interference mitigation, and so forth) on raw ADC data before first dimension FFT until point cloud detection, with the least possible intervention .
- Fast FFT computation, with programmable FFT sizes (2, 4, 8..., 2048-pt, and 3, 6, 12, ..., 768, 1536-pt) complex-input FFT and (4,8 ..., 4096-pt and 12, ..., 3072-pt) real-input FFT with improved throughput
- Operating clock frequency is 300 MHz
- Data path bit-width of 24-bits (24-bit I, 24-bit Q)
- Input and output data formatter: scaling, truncation, head/tail sample skipping, zero padding, flexible data flow and data sample arrangement to support efficient chaining of chirp processing operations and transpose accesses as required
- Capability to do all these operations in a single iteration: DC correction, interference zeroing out, complex multiplication, windowing, FFT, log, magnitude, stats computation with a capability of selecting/enabling or disabling/bypassing each of these computation blocks individually.
- Capability to do all these operations in a single iteration: DC correction, interference zeroing out, complex multiplication, windowing, FFT, with a capability of selecting/enabling or disabling/bypassing each of these computation blocks individually.
- 64KB local buffer/RAM split into eight 16KB banks, with each bank configurable as input or output for any compute/data transfer functionality.
- Flexible data flow and data sample arrangement to support efficient multi-dimensional operations and transpose accesses as required.
- Support for DC estimation and correction- Now supports 2 DC profiles
- Support for interference mitigation
  - Estimation and zeroing out of the interference samples by estimating the average of the magnitude of sudden spikes of vector elements, and applying scaled values as a threshold.
- Support for complex vector/scalar multiplications
- Complex windowing – 2K samples window RAM with a capability to split 1KB ping- 1KB pong to allow programming new coefficients while older ones are used.
- Up to 2K point FFT with efficiency of 1 sample output per cycle .
- Nested loop support in PARAM sets
- Compression engine with block floating and exponential Golomb modes with performance of 1 complex sample processing per cycle.
- Lock-step state machine
- ECC on the PARAM RAMs and parity on local buffers

## 6.2.2 Radar Accelerator Architecture

The Radar accelerator details are covered in [Chapter 24](#) , which consists of 2 parts.

The first part covers the high-level architecture and key features such as windowing, FFT, and log-magnitude. The (optional) second part covers additional features such as CFAR, complex multiplication, advanced statistics, radar data compression engine, and so forth.

## 6.3 Radar Subsystem

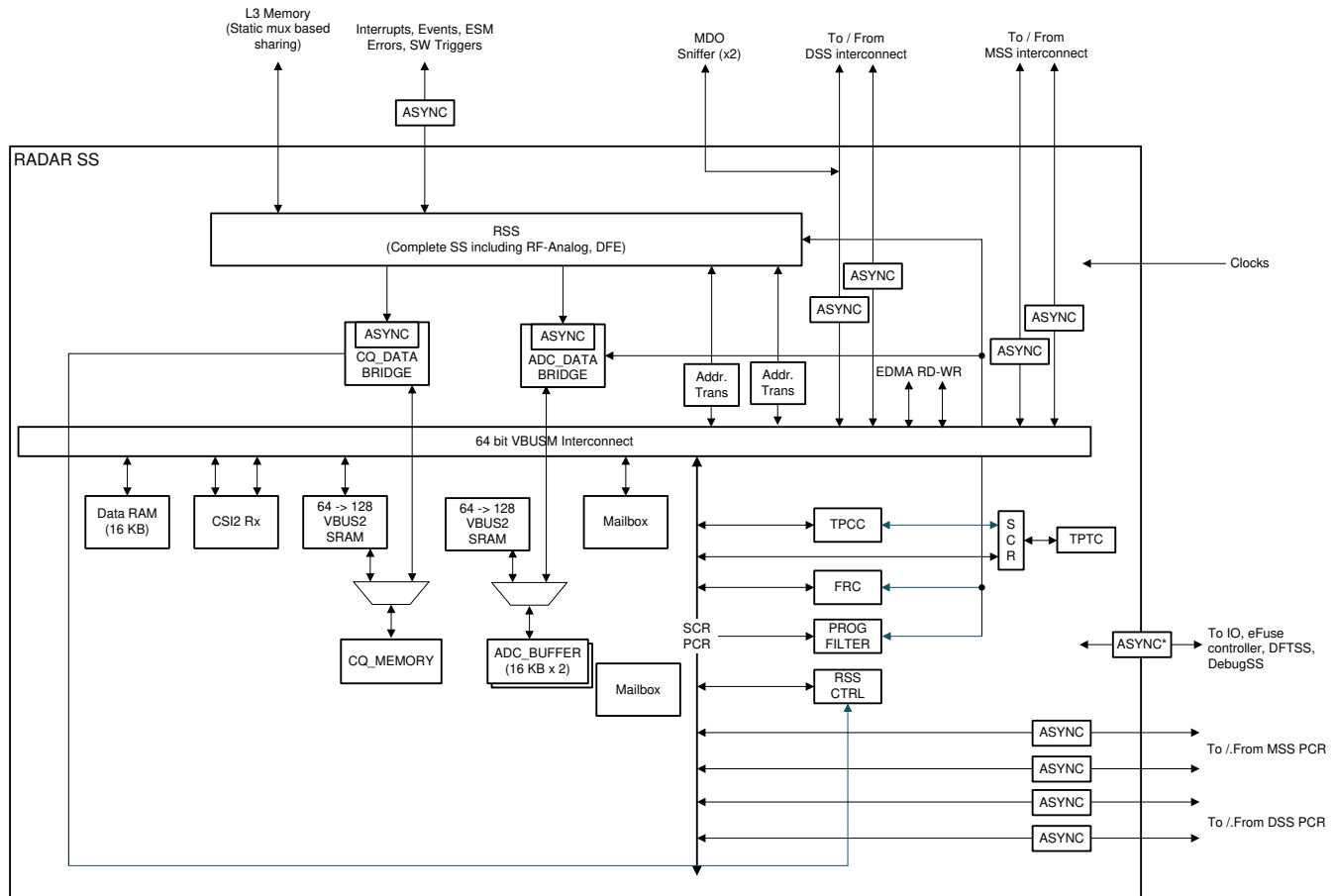
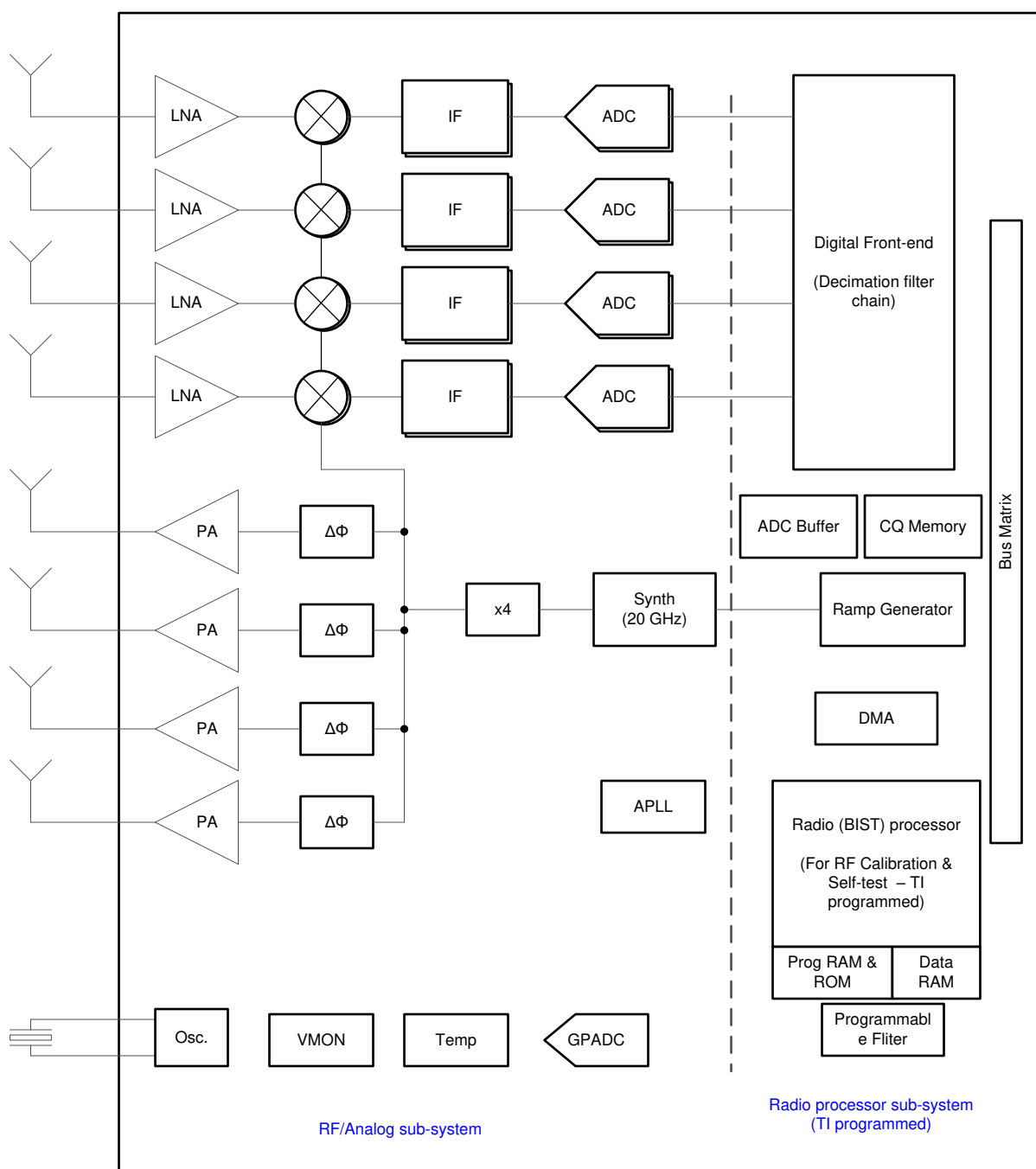


Figure 6-2. Block Diagram

### 6.3.1 BIST Subsystem

Figure 6-3 shows the BIST subsystem (BSS) block diagram.



**Figure 6-3. BIST Subsystem (BSS) Block Diagram**

The BSS consists of the RF/Analog subsystem and the radio processor subsystem.

The RF/analog subsystem implements the frequency-modulated continuous-wave transceiver system with RF and analog circuitry – namely, the synthesizer, PA, LNA, mixer, IF, and ADC. This subsystem also includes the crystal oscillator and temperature sensors. The four transmit channels can be operated up to a maximum of two at a time simultaneously. The four receive channels can all be operated simultaneously.

The radio processor subsystem includes the digital front-end, the ramp generator, and an internal processor for controlling and configuring the low-level RF/analog and ramp generator registers, based on well-defined API messages from the main subsystem. This radio processor is programmed by TI, and addresses both RF



calibration needs and some basic self-test and monitoring functions (BIST); this processor is not available directly for customer use. The digital front-end filters and decimates the raw sigma-delta ADC output, and provides the final ADC data samples at a programmable sampling rate.

BSS core is by default under reset on device power-up, SBL (secondary bootloader) or user application needs to follow below sequence to bootup BSS core

1. Prepare BSS core and memory
  - a. `RSS_CTRL.BSS_CONTROL[30:28] = 0x0`
  - b. `RSS_CTRL.BSS_CONTROL[11:0] = 0x0`
  - c. `DSS_RCM.DSS_L3_BANKD1_PD_CTRL[2:0] = 0x0`
  - d. `RSS_CTRL.BSS_TCM_MEMINIT[0] = 0x1`
  - e. wait `RSS_CTRL.BSS_TCM_MEMINIT_DONE[0]` to be value 1
  - f. wait `RSS_CTRL.BSS_TCM_MEMINIT_STATUS[0]` to be 0
  - g. `RSS_PROC_CTRL.RSS_STATIC_MEM_MEMINIT[0] = 0x1`
  - h. wait `RSS_PROC_CTRL.RSS_STATIC_MEM_MEMINIT_DONE[0]` to be value 1
2. Load the BSS firmware or Patch content from sFlash to corresponding BSS RAMs (addresses are part of FW/Patch header content).
3. Unhalt the BSS core
  - a. `MSS_CTRL.MSS_PBIIST_KEY_RST[0] = 0x0`
  - b. `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG0[31:16] = 0x0`
  - c. `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG1[31:16] = 0x0`
  - d. `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG0[15:0] = 0x28` (set XTAL Frequency 40MHz)
  - e. `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG1[15:0] = 0x28` (shadow)
  - f. `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG3[15:0] = 0x8080` (Mailbox config parameter for BSS to MSS communication)
  - g. `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG4[15:0] = same shadow value as REG3` (above register)
  - h. `RSS_CTRL.BSS_CONTROL[26-24] = 0x0`
  - i. wait `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG0` to be value 0x18F0028 (for successful bootup complete)

**Table 6-10. Description of `RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG0`**

Field Name	Bits	Description
XTAL Frequency	15:0	XTAL Frequency in MHz as an unsigned number
APLL Calibration Done	16	Expect value 1 on bootup done
APLL Calibration status	17	Expect value 1 on bootup done
BSS Boot Done	18	The bit will be set once the BSS boots up before entering the idle task loop
BSS Boot Status	19	The bit indicated that all the boot-time monitors have passed

**Table 6-10. Description of RSS\_PROC\_CTRL.RSS\_CR4\_BOOT\_INFO\_REG0 (continued)**

Table 6-10: Description of RSS\_PROC\_CTRL.RSS\_CR4\_BOOT\_INFO\_REG0 (continued)

Field Name	Bits	Description		
BSS Fault Status	22:20	Firmware Fault Status Number		
		<b>Table 6-10. Description of RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG0</b>		
		<b>Field Name</b>	<b>Bits</b>	<b>Description</b>
		0	No-Fault	
		1	BSS Firmware ASSERT	
		2	BSS Firmware CPU Abort	
		3	ESM Group 1 Error	
		4	ESM Group 2 Error	
Mailbox Boot Config Status	26:23	The set bits indicate the successful configuration of the mailbox system		
		<b>Table 6-10. Description of RSS_PROC_CTRL.RSS_CR4_BOOT_INFO_REG0</b>		
		<b>Field Name</b>	<b>Bits</b>	<b>Description</b>
		23	MSS MBOX Config success	
		26:25	RESERVED	
Reserved	31:27			

**Note**

Few set of registers from RSS\_PROC\_CTRL register space are needed for the application to read/write, those are provided in below table. Base address of RSS\_PROC\_CTRL is 0x0520 0000.

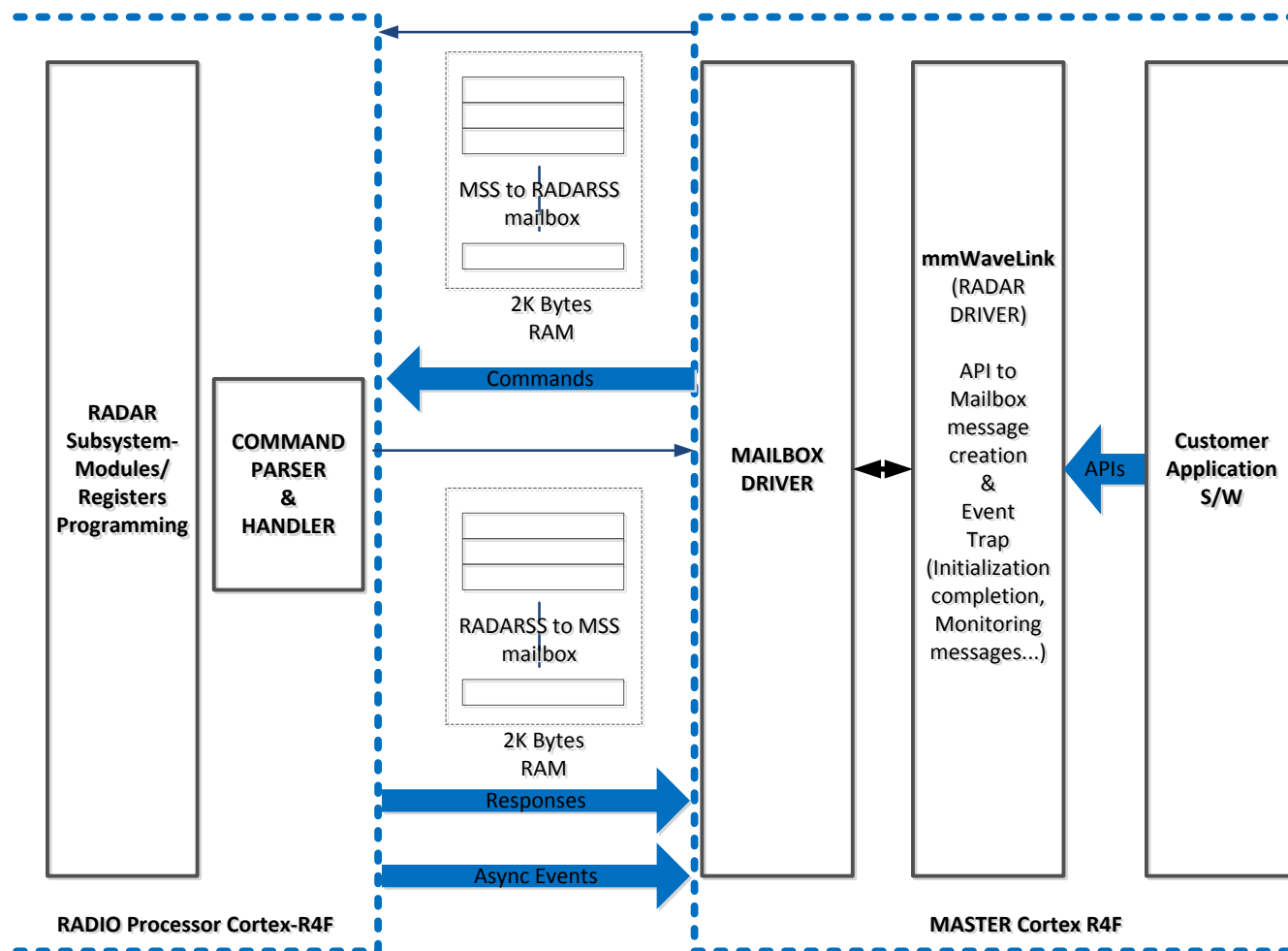
**Table 6-11. RSS\_PROC\_CTRL Registers**

Register Name	Description	Offset
RSS_CR4_BOOT_INFO_REG0	Bits defined by BSS FW, explained in above table	20h
RSS_CR4_BOOT_INFO_REG1	Shadow copy of REG0 register	24h
RSS_CR4_BOOT_INFO_REG2	Reserved	28h
RSS_CR4_BOOT_INFO_REG3	Boot time mailbox memory configuration by BSS. Bits 7:0 -> MSS Bits 15:8 -> DSS	2Ch
RSS_CR4_BOOT_INFO_REG4	Shadow copy of REG3 register	30h
RSS_CR4_BOOT_INFO_REG5	Configure BSS Logger: current configuration is Disabled. To Enable the BSS logger, set b[2:0] to 0x1. The debug data is transferred to a programmable MSS L2 memory buffer with a size of 2KB. The BSS splits this 2KB into two halves (ping and pong) to copy the debug logger data.	34h

**Table 6-11. RSS\_PROC\_CTRL Registers (continued)**

Register Name	Description	Offset
RSS_CR4_BOOT_INFO_REG6	Configure MSS L2 offset for BSS Logger. Set MSS L2 translated address offset for BSS. e.g. 0xC0260000 is the translated address of MSS L2 memory 0x10260000. 2KB of memory is needed for BSS to transfer the BSS logger data. Applications are required to make sure the 2KB starting from 0x10260000 is reserved for BSS logger.	38h
RSS_CR4_BOOT_INFO_REG7	Reserved	3Ch
RSS_STATIC_MEM_MEMINIT	Start Memory initialization of memory. Write 0x1 to start memory initialization. Write 0x0 after ensuring Memory initialization is in progress or has completed.	40h
RSS_STATIC_MEM_MEMINIT_DONE	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is complete. Write 0x1 to clear status.	44h
RSS_STATIC_MEM_MEMINIT_STATUS	Status field. Read value 0x1 indicates previously triggered Memory initialization of memory is in progress.	48h

The programming of this BSS is abstracted at a TI-defined API interface. [Figure 6-4](#) explains the high-level architecture and programming model.



**Figure 6-4. Programming Model**

The radar subsystem is accessed through a well-defined set of APIs. The application software running on the main Cortex-R5F processor, through the API call, invokes the mmWaveLink-Radar driver, which then converts these APIs into mailbox-based messages. When a message is written into the mailbox memory, an interrupt is raised to the radio processor, which then passes the message and executes the command conveyed through the mailbox message. A mailbox message read event is raised to the main R5F to indicate the radio processor has read the message.

For more details, refer to the API interface control document (ICD).

#### Note

On device bootup, RadarSS will be in the reset state. It is responsibility of SBL (secondary bootloader) or main-application (R5F), to bring RadarSS core (R4F) out of reset, load BSS firmware/patch and then un-halts (execute) it. Please refer SBL source code in MCU-PLUS-SDK-AWR2544 for the right sequence.

### 6.3.2 Clock Subsystem

The AWR2544 clock subsystem generates 76-81 GHz from an input reference of a 40/50-MHz crystal. The subsystem has a built-in oscillator circuit, followed by a clean-up PLL and an RF synthesizer circuit, as shown in Figure 6-5. The output of the RF synthesizer is then processed by an X4 multiplier to create the required frequency in the 76-81 GHz spectrum. The RF synthesizer output is modulated by the timing engine block to create the required waveforms for effective sensor operation. The timing engine is highly flexible and is

programmed through the R4F-based radio controller subsystem. For details on waveforms and programming models, refer to the Radio Control API application note.

The clean-up PLL provides a reference clock for the host processor after system wakeup.

The clock subsystem has built-in mechanisms for detecting the presence of a crystal and monitoring the quality of the generated clock.

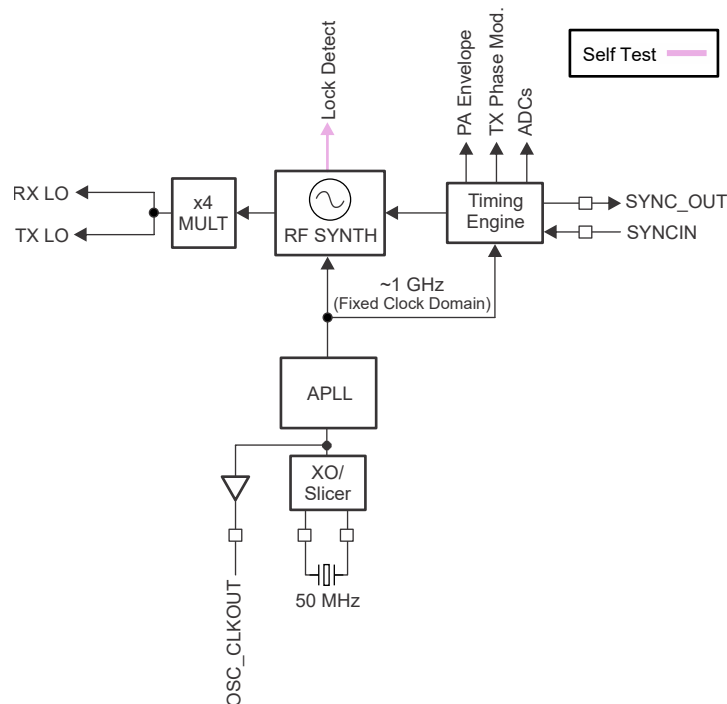


Figure 6-5. Clocking Subsystem

### 6.3.3 Transmit Subsystem

The transmit subsystem consists of four parallel transmit chains, each with independent phase and amplitude control, as shown in Figure 6-6. All transmit chains can be operational at the same time. The device supports binary phase modulation for MIMO radar and interference mitigation.

Each transmit chain is capable of delivering optimal power at the antenna port on the PCB, and supports programmable back-off for system optimization.

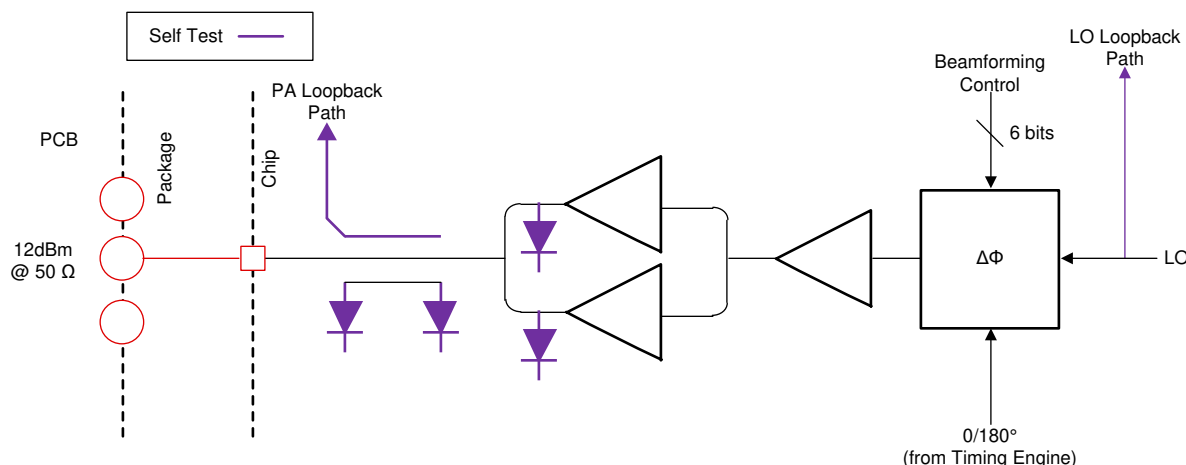


Figure 6-6. Transmit Subsystem

### 6.3.4 Receive Subsystem

The AWR2544 receive subsystem consists of four parallel channels, as shown in Figure 6-7. A single receive channel consists of an LNA, mixer, IF filtering, A2D conversion, and decimation. All four receive channels can be operational at the same time. An individual power down option is also available for system optimization.

The AWR2544 device supports an I-Only baseband architecture for each receiver channel. The AWR2544 is targeted for fast chirp systems. The bandpass IF chain has configurable lower cutoff frequencies above 350 kHz, and can support bandwidths up to 15 MHz.

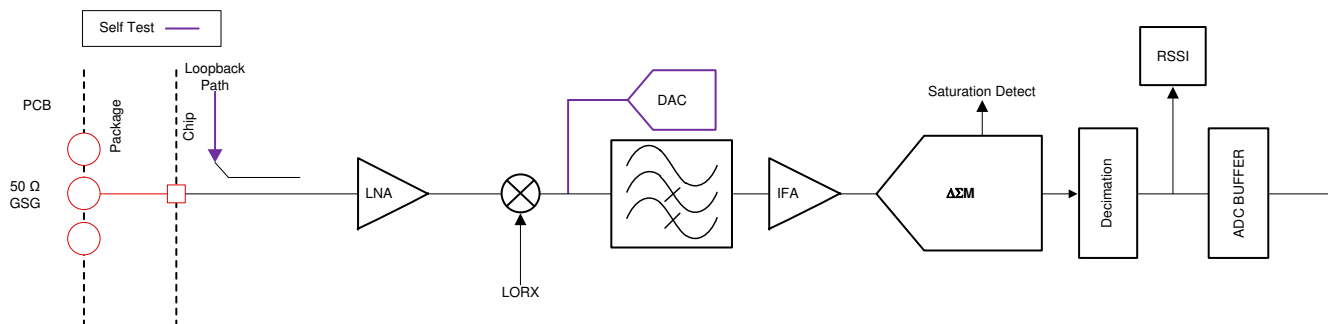
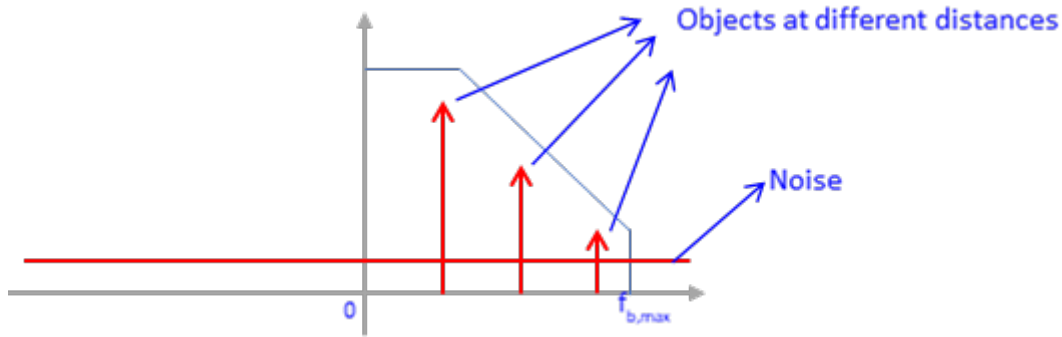


Figure 6-7. Receive Subsystem (Per Channel)

### 6.3.5 Real(I-only) Receiver in AWR2544

In FMCW radar with complex baseband (IQ) receiver, the beat frequency signals from various objects are on one-side of the complex baseband spectrum as shown in the diagram below.

Therefore, a real-only receiver chain is sufficient for FMCW radar operation, including range, velocity, and angle estimation.



**Figure 6-8. Complex Baseband Spectrum**

Although theoretically, IQ receiver can provide 3 dB SNR improvement over I-only receiver, in practice, the SNR improvement is less than 1dB for IQ over I-only receiver, after taking into consideration the IQ splitter losses. Also, with circuit optimizations, the AWR2544 device is able to achieve comparable noise-figure with real-only implementation (in comparison with IQ architecture).

There is significant power and area savings for I-only architecture, and hence it is chosen as the architecture for single chip radar device like AWR2544. Using HWA1.5, the interference detection and mitigation capability is also improved, and this enables interference-related glitches in the real ADC data to be suppressed.



This section lists the various interrupts sources supported in the different subsystem of device

### 7.1 Main Subsystem Cortex R5F Interrupt Map

**Table 7-1. Main Subsystem Cortex R5F Interrupt Map**

Interrupt Number	Define Name	Level/Pulse	Description
0	MSS_ESM_HI	Level	ESM High Priority Interrupt
1	MSS_ESM_LO	Level	ESM Low Priority Interrupt
2	CR5SS_STC_DONE	Pulse	Cortexr5ss subsystem STC complete interrupt
3	MSS_RTIA_INT0	Level	RTIA compare interrupt
4	MSS_RTIA_INT1	Level	RTIA compare interrupt
5	MSS_RTIA_INT2	Level	RTIA compare interrupt
6	MSS_RTIA_INT3	Level	RTIA compare interrupt
7	MSS_RTIA_OVERFLOW_INT0	Level	RTIA overflow interrupt
8	MSS_RTIA_OVERFLOW_INT1	Level	RTIA overflow interrupt
9	MSS_RTIB_INT0	Level	RTIB compare interrupt
10	MSS_RTIB_INT1	Level	RTIB compare interrupt
11	MSS_RTIB_OVERFLOW_INT0	Level	RTIB overflow interrupt
12	MSS_RTIB_OVERFLOW_INT1	Level	RTIB overflow interrupt
13	MSS_RTIC_INT0	Level	RTIC compare interrupt
14	MSS_RTIC_INT1	Level	RTIC compare interrupt
15	MSS_RTIC_OVERFLOW_INT0	Level	RTIC overflow interrupt
16	MSS_RTIC_OVERFLOW_INT1	Level	RTIC overflow interrupt
17	MSS_WDT_INT0	Level	WDT compare interrupt
18	MSS_WDT_INT1	Level	WDT compare interrupt
19	MSS_WDT_INT2	Level	WDT compare interrupt
20	MSS_WDT_INT3	Level	WDT compare interrupt
21	MSS_WDT_OVERFLOW_INT0	Level	WDT over flow interrupt
22	MSS_WDT_OVERFLOW_INT1	Level	WDT over flow interrupt
23	MSS_WDT_TB_INT	Level	WDT time base interrupt
24	MSS_MCRC_INT	Level	Interrupt from MCRC
25	MSS_DCCA_INT	Level	MSS_DCCA Clock compare done interrupt
26	MSS_DCCB_INT	Level	MSS_DCCB Clock compare done interrupt
27	MSS_DCCC_INT	Level	MSS_DCCC Clock compare done interrupt
28	MSS_DCCD_INT	Level	MSS_DCCD Clock compare done interrupt



**Table 7-1. Main Subsystem Cortex R5F Interrupt Map (continued)**

Interrupt Number	Define Name	Level/Pulse	Description
29	MSS_CCCA_INT	Level	MSS_CCCA dual clock compare done interrupt
30	MSS_CCCB_INT	Level	MSS_CCCB dual clock compare done interrupt
31	MSS_SPIA_INT0	Level	MSS_SPIA level 0 interrupt
32	MSS_SPIA_INT1	Level	MSS_SPIA level 1 interrupt
33	MSS_SPIB_INT0	Level	MSS_SPIB level 0 interrupt
34	MSS_SPIB_INT1	Level	MSS_SPIB level 1 interrupt
35	MSS_QSPI_INT	Pulse	MSS_QSPI interrupt
36	MSS_GIO_INT0	Level	MSS_GIO high-level Interrupt
37	MSS_GIO_INT1	Level	MSS_GIO low-level interrupt
38	MSS_ETPWMA_INT0	Level	MSS_ETPWMA Interrupt 0 which is a output to pad
39	MSS_ETPWMA_INT1	Level	MSS_ETPWMA Interrupt 1 which is a output to pad
40	MSS_ETPWMB_INT0	Level	MSS_ETPWMB Interrupt 0 which is a output to pad
41	MSS_ETPWMB_INT1	Level	MSS_ETPWMB Interrupt 1 which is a output to pad
42	MSS_ETPWC_INT0	Level	MSS_ETPWC Interrupt 0 which is a output to pad
43	MSS_ETPWC_INT1	Level	MSS_ETPWC Interrupt 1 which is a output to pad
44	MSS_CPSW_CPTS_COMP	Level	CPSW Time Stamping Trigger
45	MSS_CPTS_GENF0	Level	CPSW Generating Function 0
46	MSS_CPTS_GENF1	Level	CPSW Generating Function 1
47	MSS_CPTS_GENF2	Level	CPSW Generating Function 2
48	FHOST_EOQ_INT[0]	Level	End of Queue Interrupt form Host
49	FHOST_EOQ_INT[1]	Level	End of Queue Interrupt form Host
50	FHOST_EOQ_INT[2]	Level	End of Queue Interrupt form Host
51	FHOST_EOQ_INT[3]	Level	End of Queue Interrupt form Host
52	MSS_I2C_INT	Pulse	MSS_I2C interrupt
53	MSS_SCIA_INT0	Level	MSS_SCIA level0 input
54	MSS_SCIA_INT1	Level	MSS_SCIA level1 input
55	MSS_SCIB_INT0	Level	MSS_SCIB level0 input
56	MSS_SCIB_INT1	Level	MSS_SCIB level1 input
57	TOP_PBIST_DONE_INT	Pulse	TOP_PBIST done interrupt
58	MSS_GIO_PAD_INT0	Pulse	Interrupt Triger from GIO[0][0]
59	MSS_GIO_PAD_INT1	Pulse	Interrupt Triger from GIO[0][1]
60	MSS_GIO_PAD_INT2	Pulse	Interrupt Triger from GIO[0][2]
61	MSS_GIO_PAD_INT3	Pulse	Interrupt Triger from GIO[0][3]
62	FHOST_EOQ_INT[4]	Level	End of Queue Interrupt form Host
63	FHOST_EOQ_INT[5]	Level	End of Queue Interrupt form Host
64	FHOST_EOQ_INT[6]	Level	End of Queue Interrupt form Host
65	FHOST_EOQ_INT[7]	Level	End of Queue Interrupt form Host
66	RESERVED	RESERVED	RESERVED
67	MSS_TPCC_A_INTAGG	Level	MSS_TPCC_A Aggregated Functional Interrupt

**Table 7-1. Main Subsystem Cortex R5F Interrupt Map (continued)**

Interrupt Number	Define Name	Level/Pulse	Description
68	RESERVED	Level	
69	MSS_TPCC_A_ERRAGG	Level	MSS_TPCC_A Agregated Error Interrupt
70	RESERVED	Level	
71	MSS_GPADC_IFM_DONE	Level	MSS_GPADC ifm done interrupt
72	MSS_CPSW_TH_TRSH_INT	Pulse	MSS CPSW T-host threshold interrupt
73	MSS_CPSW_TH_INT	Pulse	MSS CPSW T-host interrupt
74	MSS_CPSW_FH_INT	Pulse	MSS CPSW F-host interrupt
75	MSS_CPSW_MISC_INT	Pulse	MSS CPSW interrupt
76	RESERVED	RESERVED	RESERVED
77	MSS_CR5A_MBOX_READ_REQ	Level	Aggregated interrupt to MSS CR5A from other processor asking it to read
78	MSS_CR5A_MBOX_READ_ACK	Level	Aggregated interrupt to MSS CR5A from other processor saying the reading from their mailbox is done.
79	RESERVED	RESERVED	RESERVED
80	RESERVED	RESERVED	RESERVED
81	RESERVED	RESERVED	RESERVED
82	TOP_DEBUGSS_TXDATA_AVAL	Pulse	Interrupt from TOP_DEBUGSS
83	MSS_CR5A_PMU_INT	Pulse	Pmu Interrupt from MSS_CR5A
84	RESERVED	RESERVED	RESERVED
85	MSS_CR5A_FPU_INT	Pulse	Floating point expection from MSS_CR5A
86	RESERVED	RESERVED	RESERVED
87	RESERVED	RESERVED	RESERVED
88	CR5A_CTI_IRQ	Level	IRQ request from CTI module from CR5A
89	RESERVED	RESERVED	RESERVED
90	RESERVED	RESERVED	RESERVED
91	MSS_SW_INT0	Pulse	Software Interrupt from MSS_CTRL
92	MSS_SW_INT1	Pulse	Software Interrupt from MSS_CTRL
93	MSS_SW_INT2	Pulse	Software Interrupt from MSS_CTRL
94	MSS_SW_INT3	Pulse	Software Interrupt from MSS_CTRL
95	MSS_SW_INT4	Pulse	Software Interrupt from MSS_CTRL
96	RESERVED	RESERVED	RESERVED
97	MSS_PERIPH_ACCESS_ERROR	Level	Aggregation of all access-errors from mpu and control spaces
98	MSS_CR5A_AHB_WR_ERR	Pulse	MSS_CR5A ahb brige getting write response as a error
99	RESERVED	RESERVED	RESERVED
100	DTHE_SHA_S_INT	Level	Interrupt form HSM_SHA

**Table 7-1. Main Subsystem Cortex R5F Interrupt Map (continued)**

Interrupt Number	Define Name	Level/Pulse	Description
101	DTHE_SHA_P_INT	Level	Interrupt form HSM_SHA
102	DTHE_TRNG_INT	Level	Interrupt form HSM_TRNG
103	DTHE_PKAE_INT	Level	Interrupt form HSM_PKAE
104	DTHE_AES_S_INT	Level	Interrupt form HSM_AES
105	DTHE_AES_P_INT	Level	Interrupt form HSM_AES
106	RESERVED	RESERVED	RESERVED
107	MSS_RTIB_INT2	Level	RTIB compare interrupt
108	MSS_RTIB_INT3	Level	RTIB compare interrupt
109	MSS_RTIC_INT2	Level	RTIC compare interrupt
110	MSS_RTIC_INT3	Level	RTIC compare interrupt
111-119	RESERVED	RESERVED	RESERVED
120	DSS_TPCC_A_INTAGG	Level	DSS_TPCC_A Aggregated Functional Interrupt
121	DSS_TPCC_A_ERRAGG	Level	DSS_TPCC_A Aggregated Error Interrupt
122-125	RESERVED	Level	RESERVED
126	DSS_PERIPH_ACCESS_ERRAGG	Level	Aggregation of access-errors from DSS peripherals. See Error access Response Section for more details
127-128	RESERVED	RESERVED	RESERVED
129	DSS_DSP_PBICT_CTRL_DONE	Pulse	DSS DSP PBICT Controller Done Interrupt
130	DSS_SW_INT0	Pulse	SW Interrupt generated by writing 0x1 to register DSS_CTRL.DSS_SW_INT[0]
131	DSS_SW_INT1	Pulse	SW Interrupt generated by writing 0x1 to register DSS_CTRL.DSS_SW_INT[1]
132	DSS_SW_INT2	Pulse	SW Interrupt generated by writing 0x1 to register DSS_CTRL.DSS_SW_INT[2]
133	DSS_SW_INT3	Pulse	SW Interrupt generated by writing 0x1 to register DSS_CTRL.DSS_SW_INT[3]
134	RESERVED	RESERVED	RESERVED
135	RESERVED	RESERVED	RESERVED
136	DSS_MCRC_INT	Level	DSS MCRC Interrupt
137-141	RESERVED	RESERVED	RESERVED
142	RSS_ADC_CAPTURE_COMPLETE	Pulse	Raw ADC capture complete interrupt from DFE-DSP bridge
143	DSS_CBUFF_INT	Pulse	DSS CBUFF Interrupt
144	DSS_CBUFF_INT_ERR	Pulse	DSS CBUFF Error Interrupt
145	DSS_HWA_THREAD1_LOOP_INT	Pulse	DSS_HWA Thread1 Loop complete interrupt
146	DSS_HWA_THREAD1_PARAM_DONE_INTR	Pulse	DSS_HWA Thread1 Param done interrupt
147	RESERVED	RESERVED	RESERVED
148	RESERVED	RESERVED	RESERVED
149	RESERVED	RESERVED	RESERVED

**Table 7-1. Main Subsystem Cortex R5F Interrupt Map (continued)**

Interrupt Number	Define Name	Level/Pulse	Description
150	DFE_FRAME_START_TO_DSS	Pulse	Frame start interrupt from BSS.
151	RSS_TPCC_A_INTAGG	Level	RSS_TPCC_A Aggregated Functional Interrupt
152	RSS_TPCC_A_ERRAGG	Level	RSS_TPCC_A Aggregated Error Interrupt
153	ADC_VALID_FALL_EDGE	Pulse	Interrupt is trigger during Falling edge of ADC valid
154	DFE_FRAME_START_TO_MSS	Pulse	Frame start interrupt from RSS which can be masked by RSS.
155	DFE_CHIRP_CYCLE_START	Pulse	Chirp cycle start interrupt from DFE
156	DFE_CHIRP_CYCLE_END	Pulse	Chirp cycle end interrupt from DFE
157	DFE_END_OF_FRAME	Pulse	End of Frame interrupt from DFE
158	RSS_FRC_FRAME_START	Pulse	Frame start interrupt from FRC
159	RSS_ADC_CAPTURE_COMPLETE_DITH	Pulse	ADC capture complete interrupt from DFE-DSP bridge after dithering
160	DSS_HWA_THREAD2_LOOP_INTERRUPT	Pulse	DSS_HWA Thread2 Loop complete interrupt
161	DSS_HWA_THREAD2_PARAM_DONE_INT	Pulse	DSS_HWA Thread2 Param done interrupt
162	DSS_WDT_TB_INT	Level	DSS WDT Time Base Interrupt
163	DSS_HWA_LOCAL_RAM_ERR	Level	DSS_HWA Local RAM access error
164-187	RESERVED	Level	RESERVED
188	DSS_SW_INT_RSS_PROC_CTRL0	Pulse	Interrupt from RSS_PROC_CTRL
189	DSS_SW_INT_RSS_PROC_CTRL1	Pulse	Interrupt from RSS_PROC_CTRL
190	DSS_SW_INT_RSS_PROC_CTRL2	Pulse	Interrupt from RSS_PROC_CTRL
191	DSS_SW_INT_RSS_PROC_CTRL3	Pulse	Interrupt from RSS_PROC_CTRL
192	RSS_DATA_CAPTURE_ENABLE_FALL	Pulse	Interrupt is triggered data_capture enable fall from DFE-DSP bridge (toggles after completion of every chirp)
193-206	RESERVED	RESERVED	RESERVED
207	RSS_EXT_SYNC_OUT	Pulse	rss_int_map[29]
208	RSS_EXT_SYNC_IN	Pulse	rss_int_map[30]
209-218	RESERVED	RESERVED	RESERVED
219	RSS_BSS_CR4_STC_COMPLETE	Pulse	CR4 STC test complete
220	RSS_FRC_EVT_GEN_1	Pulse	rss_int_map[51]
221	RSS_FRC_EVT_GEN_2	Pulse	rss_int_map[52]
222	RSS_FRC_EVT_GEN_3	Pulse	rss_int_map[53]
223	RSS_FRC_EVT_GEN_4	Pulse	rss_int_map[54]
224	RSS_FRC_TS_1_TRIG	Pulse	rss_int_map[55]
225	RSS_FRC_TS_2_TRIG	Pulse	rss_int_map[56]
226	RSS_FRC_TS_3_TRIG	Pulse	rss_int_map[57]

**Table 7-1. Main Subsystem Cortex R5F Interrupt Map (continued)**

Interrupt Number	Define Name	Level/Pulse	Description
227	RSS_FRC_TS_4_TRIG	Pulse	rss_int_map[58]
228	RESERVED	RESERVED	RESERVED
229	MSS_GIO_PAD_INT4	Pulse	Interrupt Triger from MSS GIO[1][0]
230	MSS_GIO_PAD_INT5	Pulse	Interrupt Triger from MSS GIO[1][1]
231	MSS_GIO_PAD_INT6	Pulse	Interrupt Triger from MSS GIO[1][2]
232	MSS_GIO_PAD_INT7	Pulse	Interrupt Triger from MSS GIO[1][3]
233-235	RESERVED	RESERVED	RESERVED
236	DSS_CSI_INT	Level	DSS_CSITX Aggregated Interrupt form the IP
237	DSS_CSI_PARITY_ERROR	Level	Parity Error form DSS_CSITX
238-255	RESERVED	RESERVED	RESERVED

## 7.2 Main Subsystem ESM Interrupt Map

**Table 7-2. Main Subsystem ESM Interrupt Map Group 1**

ESM GROUP1	Define Name	Description
127:125	RESERVED	RESERVED
124	DSS_CSI_VBUSP2OCP_ACCESS_ERR	CSI access error when an SOC master is trying to access DSS_CSI_TX
123	ANA_WU_AND_CLK_STATUS_ERR	Aggregated Error from ANA_WU_STATUS_REG and ANA_CLK_STATUS_REG
122	MSS_BUS_SAFETY_CR5B_AHB	AHB bridges safety Error for Cr5B - Comparision Error of all outputs from AHB bridge of CR5A
121	MSS_BUS_SAFETY_CR5A_AHB	AHB bridges safety Error for Cr5A - Comparision Error of all outputs from AHB bridge of CR5A
120-112	RESERVED	RESERVED
111	MSS_BUS_SAFETY_MSS2R5SS	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
110	MSS_BUS_SAFETY_DSS2R5SS	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
109	MSS_BUS_SAFETY_R5SS2MSS	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
108	MSS_BUS_SAFETY_R5SS2DSS	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
107	DSS_TPCC_A_ERRAGG	
106	DSS_CBUFF_SBE_ERR	DSS_CBUFF FIFO Single Bit error

**Table 7-2. Main Subsystem ESM Interrupt Map Group 1 (continued)**

ESM GROUP1	Define Name	Description
105	DSS_CBUFF_DBE_ERR	DSS_CBUFF FIFO Double Bit error
104	DSS_CBUFF_SAFETY_ERR	DSS_CBUFF FIFO Safety error
103	MPU_DSS_L3_BANKA_ADDR_ERR	MPU Addressing Error for DSS_(X)_MPU_ADDR_ERR
102	MPU_DSS_L3_BANKB_ADDR_ERR	MPU Addressing Error for DSS_(X)_MPU_ADDR_ERR
101	MPU_DSS_L3_BANKA_PROT_ERR	MPU Addressing Error for DSS_(X)_MPU_PROT_ERR
100	MPU_DSS_L3_BANKB_PROT_ERR	MPU Addressing Error for DSS_(X)_MPU_PROT_ERR
99	DSS_ECC_AGGR_UERR	DSS ECC AGGR Un-Correctable Error - DSS_L3_BANKA/B - DSS_TPTC_A0/1 FIFO
98	DSS_ECC_AGGR_SERR	DSS ECC AGGR Correctable Error - DSS_L3_BANKA/B - DSS_TPTC_A0/1 FIFO
97	DSS_HWA_GRP1/2_ERR	or of HWA group1 or group2 error
96	DSS_L3_BANKA_ECC_UERR	DSS_L3_BANKA Uncorrectable ECC Error
95	DSS_L3_BANKB_ECC_UERR	DSS_L3_BANKB Uncorrectable ECC Error
94	DSS_L3BANKA_ACCESS_ERR	
93	DSS_L3BANKB_ACCESS_ERR	
92	CSI_TX_PARITY_ERROR	Parity ERROR from CSI TX Buffers
91	MSS_BUS_SAFETY_CR5A_SLV	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
89 to 90	RESERVED	RESERVED
88	MSS_MPU_MSS2DSS_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
87	MSS_MPU_MSS2DSS_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
86	MSS_MPU_MBOX_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
85	MSS_MPU_MBOX_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
84	MSS_MPU_L2_BANKA_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
83	MSS_MPU_L2_BANKA_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
82	MSS_MPU_L2_BANKB_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
81	MSS_MPU_L2_BANKB_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
80	MSS_MPU_DTHE_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
79	MSS_MPU_PCRA_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
78	MSS_MPU_QSPI_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
77	MSS_MPU_CR5A_SLV_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
76	MSS_MPU_L2_BANKC_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR

**Table 7-2. Main Subsystem ESM Interrupt Map Group 1 (continued)**

ESM GROUP1	Define Name	Description
75	MSS_MPU_HSM_SLV_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
74	MSS_MPU_DTHE_PROT_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
73	MSS_MPU_PCRA_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
72	MSS_MPU_QSPI_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
71	MSS_MPU_CR5A_SLV_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
70	MSS_MPU_L2_BANKC_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
69	MSS_MPU_HSM_SLV_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
68	MSS_CPSW_SERR	Cpsw memories Single bit error pulse
67	MSS_CPSW_UERR	Cpsw memories Double bit error pulse
66	MSS_BUS_SAFETY_SEC_AGG_ERR	Aggregated error for SEC from all Nodes in MSS_SCR
64 to 65	RESERVED	RESERVED
63	ANA_LIMP_MODE	Error signal from analog if the CLK monitor finds the REF CLK to be outside the permissible range of frequency
62	MSS_MCRC_ERR	MCRC Comparision Error
61	MSS_DCCA_ERR	DCCA frequency comparision error
60	MSS_DCCB_ERR	DCCB frequency comparision error
59	MSS_DCCC_ERR	DCCC frequency comparision error
58	MSS_DCCD_ERR	DCCD frequency comparision error
57	MSS_CCCA_ERR	CCCA frequency comparision error
56	MSS_CCCB_ERR	CCCB frequency comparision error
55	MSS_SPIA_SERR	Single Bit correctable error indication for MIBSPI-A multi-buffer
54	MSS_SPIB_SERR	Single Bit correctable error indication for MIBSPI-B multi-buffer
53	MSS_SPIA_UERR	Multi Bit uncorrectable error indication for MIBSPI-A muti-buffer
52	MSS_SPIB_UERR	Multi Bit uncorrectable error indication for MIBSPI-B muti-buffer
51	RSS_TPCC_A_ERRAGG	RSS_TPCC_A Aggregated Error Interrupt - TPCC Error - TPCC MPU Error - TPTC Error for all TPTCs connected to TPCC - Read and Write Config Space Access error to TPCC - Read and Write Config Space Access error or all TPTCs connected to TPCC
50	RSS_ECC_AGG_SERR	SEC error from ECC-AGGREGATOR which controls ADC_BUF_Memories and TPTC-memories
49	RSS_ECC_AGG_UERR	DED error from ECC-AGGREGATOR which controls ADC_BUF_Memories and TPTC-memories
48-45	RESERVED	RESERVED
44	MSS_TCMA_CR5A_SERR	Single Bit correctable error indication for ATCM of CR5A

**Table 7-2. Main Subsystem ESM Interrupt Map Group 1 (continued)**

ESM GROUP1	Define Name	Description
43	MSS_TCMB1_CR5A_SERR	Single Bit correctable error indication for B1TCM of CR5A
42	MSS_TCMB0_CR5A_SERR	Single Bit correctable error indication for B0TCM of CR5A
41-39	RESERVED	RESERVED
38	MSS_CR5A_ITAG_SERR	Single Bit correctable error indication for Cache ITAG of CR5A
37	MSS_CR5A_IDATA_SERR	Single Bit correctable error indication for Cache IDATA of CR5A
36	MSS_CR5A_DTAG_SERR	Single Bit correctable error indication for Cache DTAG of CR5A
35	MSS_CR5A_DDATA_SERR	Single Bit correctable error indication for Cache DDATA of CR5A
34-31	RESERVED	RESERVED
30	MSS_TPCC_A_AGG_ERR	MSS_TPCC_A Aggregated Error Interrupt
		- TPCC Error
		- TPCC MPU Error
		- TPTC Error for all TPTCs connected to TPCC
		- Read and Write Config Space Access error to TPCC
		- Read and Write Config Space Access error or all TPTCs connected to TPCC
29-28	RESERVED	RESERVED
27	EFUSE_ERR	Reserved for efuse errors
26	MSS_STC_ERR	STC Error indication for MSS Cortex5ss
25	MSS_CCMR5_ST_ERR	CORTEXR5-Sub System Self test error for CCMR5 (comparator module)
24	RESERVED	RESERVED
23	QSPI_WR_ERR	QSPI write error
22	MSS_ECC_AGGR_CR5A_SERR	MSS ECC AGGR for CR5A Memories Correctable Error
		- MSS_VIM_CR5A
		- MSS_CR5A_CACHES (only for injection. Error is sent out through event bus)
		- MSS_CR5A_TCMs (only for injection. Error is sent out through event bus)
21-19	RESERVED	RESERVED
18	MSS_ECC_AGGR_SERR	MSS ECC AGGR Correctable Error
		- MSS_L2_BANKA/B
		- MSS_MBOX
		- MSS_RETRAM
		- MSS_GPADC
		- MSS_TPTC_A0/1 FIFO
		- MSS_TPTC_B0 FIFO



**Table 7-2. Main Subsystem ESM Interrupt Map Group 1 (continued)**

ESM GROUP1	Define Name	Description
17	MSS_ECC_AGGR_UERR	MSS ECC AGGR Un-Correctable Error
		- MSS_L2_BANKA/B
		- MSS_MBOX
		- MSS_RETRAM
		- MSS_GPADC
		- MSS_TPTC_A0/1 FIFO
		- MSS_TPTC_B0 FIFO
16	MSS_L2_BANKC_ACC_ERR	MSS_L2_BANKC access Error when accessed during power down
15	MSS_L2_BANKB_ACC_ERR	MSS_L2_BANKB access Error when accessed during power down
14	MSS_L2_BANKA_ACC_ERR	MSS_L2_BANKA access Error when accessed during power down
13-4	RESERVED	RESERVED
3	BSS_ESM_LO	ESM IRQ from BSS
2	BSS_ESM_HI	ESM FIQ from BSS
1	HSM_ESM_LO	ESM IRQ from HSM
0	HSM_ESM_HI	ESM FIQ from HSM

**Table 7-3. Main Subsystem ESM Interrupt Map Group 2**

ESM GROUP2	Define Name	Description
31 to 30	RESERVED	RESERVED
29	MSS_BUS_SAFETY_L2_BANKC	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
28	MSS_L2_BANKC_ECC_UERR	MSS_L2_BANKC Uncorrectable ECC Error
27	MSS_DCCA_ERR	DCCA Frequency comparison error
26	BSS_ESM_HI	ESM FIQ from BSS
25	ANA_WU_AND_CLK_STATUS_ERR	Aggregated Error from ANA_WU_STATUS_REG and ANA_CLK_STATUS_REG
24	MSS_BUS_SAFETY_CR5A_MST_RD	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
23	MSS_BUS_SAFETY_CR5B_MST_RD	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
22	MSS_BUS_SAFETY_CR5A_MST_WR	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
21	MSS_BUS_SAFETY_CR5B_MST_WR	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control

**Table 7-3. Main Subsystem ESM Interrupt Map Group 2 (continued)**

ESM GROUP2	Define Name	Description
20	MSS_BUS_SAFETY_L2_BANKA	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
19	MSS_BUS_SAFETY_L2_BANKB	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
18	MSS_BUS_SAFETY_PCRA	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
17	MSS_BUS_SAFETY_PCRB	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X)
		- ECC Uncorrectable Data Error
		- Compare Error on Control
16	MSS_ECC_AGGR_CR5A_UERR	MSS ECC AGGR for CR5A Memories Un-Correctable Error
		- MSS_VIM_CR5A
		- MSS_CR5A_CACHES (only for injection. Error is sent out through event bus)
		- MSS_CR5A_TCMs (only for injection. Error is sent out through event bus)
15	MSS_ECC_AGGR_CR5B_UERR	MSS ECC AGGR for CR5A Memories Un-Correctable Error
		- MSS_VIM_CR5A
		- MSS_CR5A_CACHES (only for injection. Error is sent out through event bus)
		- MSS_CR5A_TCMs (only for injection. Error is sent out through event bus)
14	MSS_L2_BANKA_ECC_UERR	MSS_L2_BANKA Uncorrectable ECC Error
13	MSS_L2_BANKB_ECC_UERR	MSS_L2_BANKB Uncorrectable ECC Error
12	VIM_LOCK_ERR	MSS_VIM lock step compare error
11	MSS_WDT_NMI	MSS Watch dog timer non maskable irq
10	MSS_CR5A_LIVELOCK	MSS_CR5A in live lock due to fatal errors
9	MSS_CR5B_LIVELOCK	MSS_CR5B in live lock due to fatal errors
8	MSS_TCMB1_CR5B_PARITY_ERR	Parity Error on Control signals for B1TCM of CR5B
7	MSS_TCMB0_CR5B_PARITY_ERR	Parity Error on Control signals for B0TCM of CR5B
6	MSS_TCMA_CR5B_PARITY_ERR	Parity Error on Control signals for ATCM of CR5B
5	MSS_TCMB1_CR5A_PARITY_ERR	Parity Error on Control signals for B1TCM of CR5A
4	MSS_TCMB0_CR5A_PARITY_ERR	Parity Error on Control signals for B0TCM of CR5A
3	MSS_TCMA_CR5A_PARITY_ERR	Parity Error on Control signals for ATCM of CR5A
2	MSS_CCMR5_ERR	Lock step Comparison Error from CCMR5
1	DSS_ESM_HI	ESM FIQ from DSP

**Table 7-3. Main Subsystem ESM Interrupt Map Group 2 (continued)**

ESM GROUP2	Define Name	Description
0	HSM_ESM_HI	ESM FIQ from HSM

**Table 7-4. MSS R5F ESM Interrupt Map**

ESM GROUP3	Define Name	Description
31	RESERVED	RESERVED
30	RESERVED	RESERVED
29	MSS_CR5B_DDATA_UERR	Dcache data multibit error from CR5B
28	RESERVED	RESERVED
27	MSS_CR5B_DTAG_UERR	Dcache tag multibit error from CR5B
26	RESERVED	RESERVED
25	RESERVED	RESERVED
24	RESERVED	RESERVED
23	RESERVED	RESERVED
22	RESERVED	RESERVED
21	MSS_CR5A_DDATA_UERR	Dcache data multibit error from CR5A
20	RESERVED	RESERVED
19	MSS_CR5A_DTAG_UERR	Dcache tag multibit error from CR5A
18	RESERVED	RESERVED
17	RESERVED	RESERVED
16	RESERVED	RESERVED
15	RESERVED	RESERVED
14	RESERVED	RESERVED
13	MSS_TCMA_CR5B_UERR	Multi Bit Error in ATCM of CR5B
12	RESERVED	RESERVED
11	MSS_TCMB1_CR5B_UERR	Multi Bit Error in B1TCM of CR5B
10	RESERVED	RESERVED
9	MSS_TCMB0_CR5B_UERR	Multi Bit Error in B0TCM of CR5B
8	RESERVED	RESERVED
7	MSS_TCMA_CR5A_UERR	Multi Bit Error in ATCM of CR5A
6	RESERVED	RESERVED
5	MSS_TCMB1_CR5A_UERR	Multi Bit Error in B1TCM of CR5A
4	RESERVED	RESERVED
3	MSS_TCMB0_CR5A_UERR	Multi Bit Error in B0TCM of CR5A
2	RESERVED	RESERVED
1	EFUSE_AUTOLOAD_ERR	Reserved for efuse autoloader error
0	RESERVED	RESERVED



The ADC buffer is on-chip memory arranged as a ping-pong buffer, with ECC support for each ping and pong memory. The raw ADC output data from RADAR-SS is stored on this memory, to be consumed by the DSP, or by the hardware FFT accelerator for the post processing.

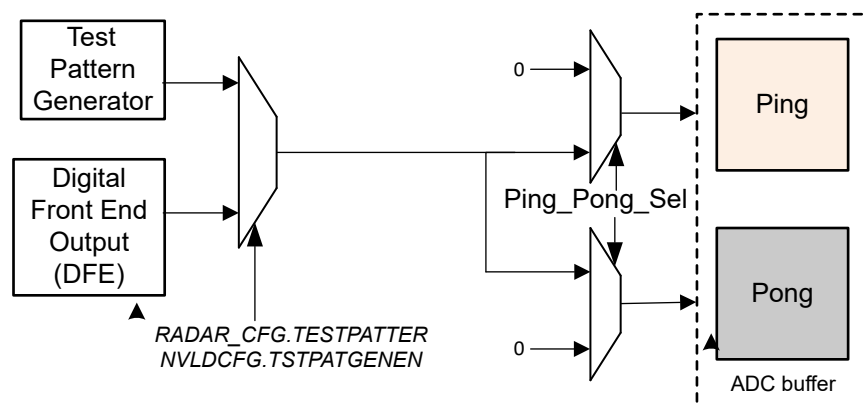
For the application software, the ADC buffer (either ping or pong) is seen as a single memory at the base address.

RadarSS generates Chirp Parameter (CP) and Chirp Quality (CQ) data along with ADC data. Interface Control Document (ICD) explains CP/CQ data format and steps to enable this feature. Refer ICD for the latest supported features towards CP/CQ/ADC of the device.

## 8.1 Functional Description

Figure 8-1 shows the block diagram of the ADC buffer scheme. The two data input sources to the ADC buffer are:

- Raw ADC output data from the digital front end (DFE)
- Ramp pattern data from the test pattern generator



**Figure 8-1. ADC Buffer Block Diagram**

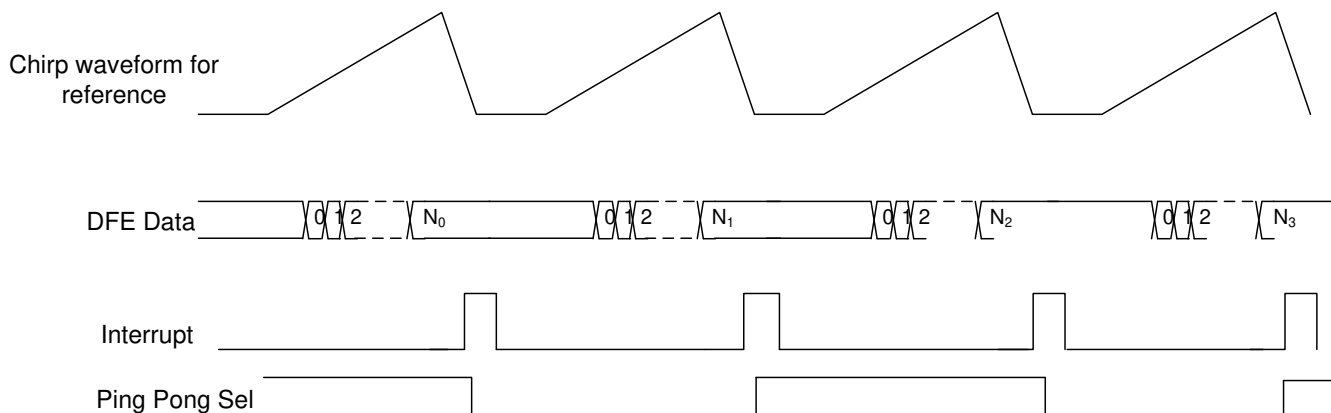
### 8.1.1 DFE Data Write Operation

The ADC buffer can be written from DFE in any of the three modes by configuring the control registers ADCBUFCFG1, ADCBUFCFG2, ADCBUFCFG3, and ADCBUFCFG4 in APP\_HWA\_ADCBUF\_CTRL address space:

1. Single-chirp mode
2. Multi-chirp mode
3. Continuous mode

The DFE data from the three Rx channels can be independently enabled by programming the register ADCBUFCFG1.

In single-chirp mode, the FMCW chirp data from the DFE is written to the ADC buffer on a per chirp basis, and a chirp available interrupt is generated on the completion of the write data operation at the end of the chirp, as shown in Figure 8-2. ADC buffer control logic generates the Ping\_Pong\_Sel signal, as shown in Figure 8-2, which controls whether the data is written into either ping or pong buffer. Data write can start from either the ping or pong buffer.



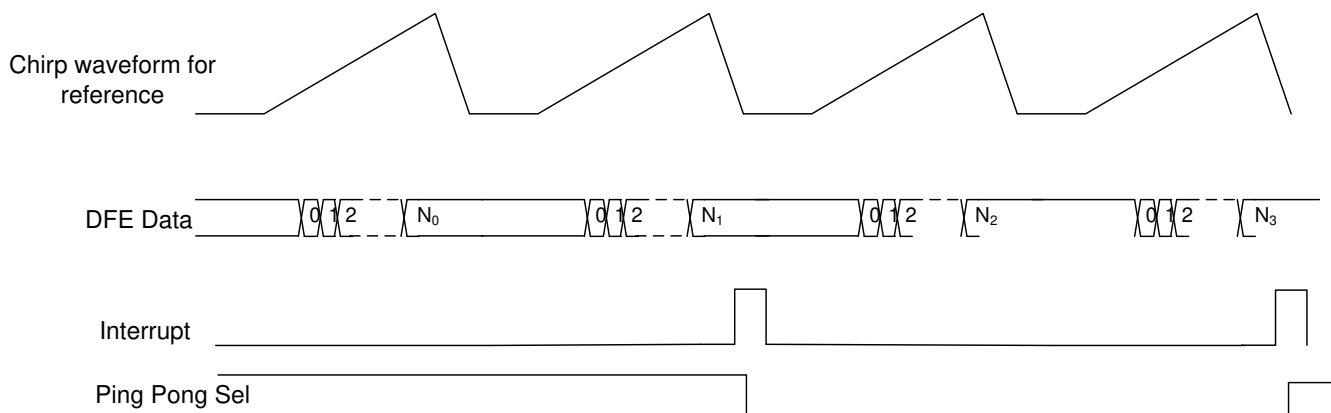
**Figure 8-2. Single-Chirp Mode**

In multi-chirp mode, ADC samples for N chirps are stored in a ping/pong buffer before the Ping Pong Select toggles and the Chirp Available Interrupt is generated. The number of chirps stored in the ping and the pong buffer are configured in the register field ADCBUFNUMCHRPING.

ADC Buffer Single-Chirp and Multi-Chirp Mode Programming Sequence shows the programming sequence for the ADC buffer single-chirp and multi-chirp modes.

#### Note

Registers for ping and pong must be programmed with the same value for correct functionality.



**Figure 8-3. Multi-Chirp Mode**

In continuous mode, where the FMCW transceiver is configured to output a single frequency tone in the range of X-Y GHz (where X is start and Y is end frequency supported by the device), 'N' ADC samples are stored in a ping/pong buffer before the Ping Pong Select toggles and the Chirp Available Interrupt is generated. The value N is configured in the field APP\_HWA\_ADCBUF\_CTRL.ADCBUFCFG4.ADCBUFSAMPCNT. In real mode, this value N refers to the number of real samples per channel, and in complex mode, this refers to the number of complex samples per channel. This counter increments once for every new sample (as long as 1 or more Rx channels are enabled). Continuous mode is expected to be only used for CZ and ADC buffer testpattern mode.

ADC Buffer Continuous Mode Programming Sequence shows the programming sequence for ADC buffer continuous mode.

### 8.1.2 Test Pattern Generator Support

An internal test pattern generator which outputs a ramp pattern helps during the initial software development and debug. The output of this module is muxed with the DFE data before sending it to the ADC buffers, as shown in [Figure 8-1](#). Because this is meant for testing the path from the ADC buffer until the final output through LVDS, the ADC buffer configurations must be set to continuous streaming mode, in which the ping-pong switch is based on the number of samples. The test pattern generator can be configured by programming the register TESTPATTERNVLDCFG in the RADAR\_CFG address space. Additional configurable registers are provided for configuring the ramp pattern output from the test pattern generator, such as offset at the start of ramp, step size, and so forth. Refer to the RSS\_CTRL address space and test pattern generator-related registers for further information.

### 8.1.3 ADC Buffer Data Formats

The data is written in the following formats to the ADC buffer:

- Non-interleaved data format

#### 8.1.3.1 Non-Interleaved Data Format

In non-interleaved mode storage, each channel data is stored in different memory locations, as shown in [Table 8-1](#).

**Table 8-1. Non-Interleaved Data Format**

RX0(3)	RX0(2)	RX0(1)	RX0(0)
RX0(7)	RX0(6)	RX0(5)	RX0(4)
RX1(3)	RX1(2)	RX1(1)	RX1(0)
RX1(7)	RX1(6)	RX1(5)	RX1(4)
RX2(3)	RX2(2)	RX2(1)	RX2(0)
RX2(7)	RX2(6)	RX2(5)	RX2(4)

## Chapter 9

# Ethernet SubSystem (MCU\_CPSW0)

---



This chapter describes the Ethernet MAC (Media Access Controller). For conceptual purposes the below documentation refers to this MAC as being a two port CPSW with port 0 being the CPPI DMA host port and port 1 being the Ethernet port.

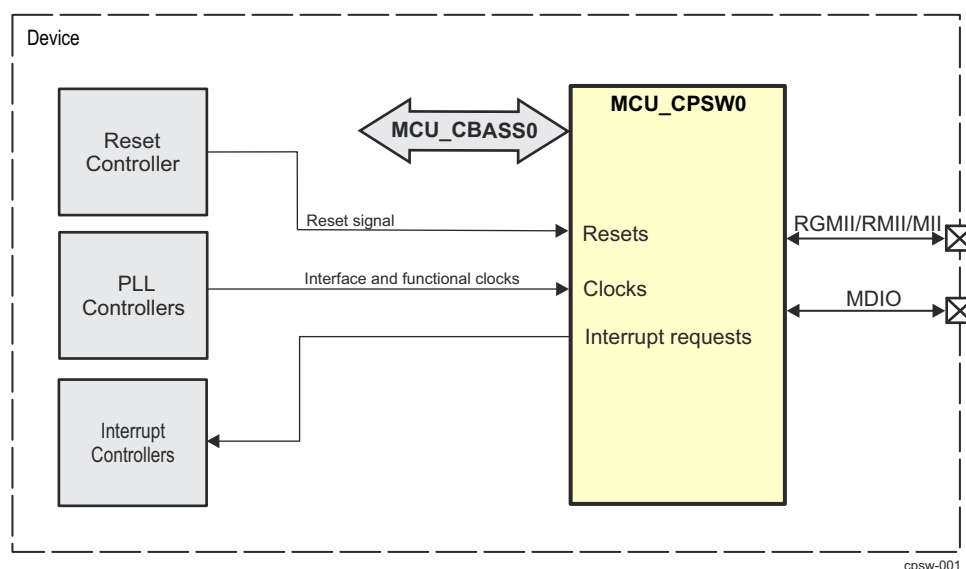
9.1 MCU_CPSW0 Overview.....	2546
9.2 MCU_CPSW0 Environment.....	2549
9.3 MCU_CPSW0 Integration.....	2552
9.4 MCU_CPSW0 Functional Description.....	2554
9.5 MCU_CPSW0 Programming Guide.....	2649
9.6 MSS_CPSW Registers.....	2652

## 9.1 MCU\_CPSW0 Overview

The two-port Ethernet MAC (MCU\_CPSW0) subsystem provides Ethernet packet communication for the device and is configured in a similar manner as a two-port Ethernet switch. MCU\_CPSW0 features the Reduced Gigabit Media Independent Interface (RGMI), Reduced Media Independent Interface (RMII), and the Management Data Input/Output (MDIO) interface for physical layer device (PHY) management.

The device has integrated two-port Ethernet Switch subsystem into device MCU domain named MCU\_CPSW0.

Figure 9-1 shows the MCU\_CPSW0 module overview.



**Figure 9-1. MCU\_CPSW0 Overview**

### 9.1.1 MCU\_CPSW0 Features

The MCU\_CPSW0 subsystem provides the following features:

- One Ethernet port with selectable MII, RMII and RGMI interfaces (port 1).
- One CPDMA CPPI DMA Host Interface (Port 0)
- Synchronous 1 Gbps operation.
- Flexible logical FIFO-based packet buffer structure
- Eight priority level Quality Of Service (QOS) support (802.1p)
- Support for Audio/Video Bridging (P802.1Qav/D6.0) (**C**redit **B**ased **S**haper)
- Support for IEEE 1588 Clock Synchronization (2008 Annex D, Annex E and Annex F)
  - Timestamp module capable of time stamping external timesync events like Pulse-Per-Second and also generating Pulse-Per-Second outputs
  - CPTS module that supports time stamping for IEEE1588 with support for 4 hardware push events and generation of compare output pulses
- DSCP Priority Mapping (IPv4 and IPv6)
- Energy Efficient Ethernet (EEE) support (802.3az)
- Flow Control (802.3x) Support
- Non Blocking switch fabric
- Time Sensitive Network Support
  - IEEE 802.1Qbv/D2.2 Enhancements for Scheduled Traffic
- Address Lookup Engine (ALE)
  - Configurable number of addresses plus VLANs
  - Wire rate lookup
  - Host controlled time-based aging and/or auto-aging
  - Spanning tree support



- L2 address lock and L2 filtering support
- MAC authentication (802.1x)
- Receive-based or destination-based Multicast and Broadcast rate limits
- MAC address blocking
- Source port locking
- OUI (Vendor ID) host accept/deny feature
- Configurable number of classifier/policers
- VLAN support
  - 802.1Q compliant
    - Auto add port VLAN for untagged frames on ingress
    - Auto VLAN removal on egress and with pad to minimum frame size
- EtherStats and 802.3Stats Remote network Monitoring (RMON) statistics gathering (per port)
- Ethernet or Castagnoli CRC selectable on Ethernet egress
- Digital Loopback supported (Ethernet egress to Ethernet ingress)
- OAM Loopback supported (FIFO Loopback - port ingress to egress including host port)
- CPSPGMII Loopback Modes (Ethernet egress to Ethernet ingress)
- Maximum frame size of 2024 bytes (including VLAN)
- Management Data Input/Output (MDIO) module for PHY Management with Clause 45 support
- Programmable interrupt control with selected interrupt pacing
- Emulation support
- Full duplex mode supported in 10/100 Mbps. Half-duplex mode supported only in 10/100 Mbps modes only.
- RAM Error Detection and Correction (SECDED)
- InterVLAN Routing is supported - 4 routes per egress port.
- Automotive Security Features
  - VLANs can be configured to not allow fragmented IPv4 frames (that is, fragmented IPv4 traffic)
  - VLANs can be configured to only allow up to four different IPv4 Protocols or IPv6 Next Header values
  - Drop invalid source addresses, that is drop Source Addresses with bit 40 set (Multicast/Broadcast indicator on Destination Addresses)
  - Drop frames that the IEEE802.3 length is not contained within the frame. (Ether Types 0-1500)
  - Any source address can be secured to a port dropping any attempts from other ports to masquerade as a service
  - Any source or destination address can be blocked
  - Per port or per VLAN ingress checking, dropping traffic from non-member ports
  - Classification, policing on L2 and L3 information

### 9.1.2 MCU\_CPSW0 Not Supported Features

The following MCU\_CPSW0 features are not supported:

- GMII Mode
- SGMII Mode
- MACSEC
- Synchronous Ethernet
- Software reset
- Rate-limiting is not supported in half-duplex mode
- Dual VLAN switch operations are not supported
- RGMII Internal Delay Mode disabled.
- Priority based flow control is not supported.
- InterVLAN routed packets will be dropped if the FIFO room is insufficient regardless of receive 803.3x flow control.

### 9.1.3 Terminology

**Terminology:**

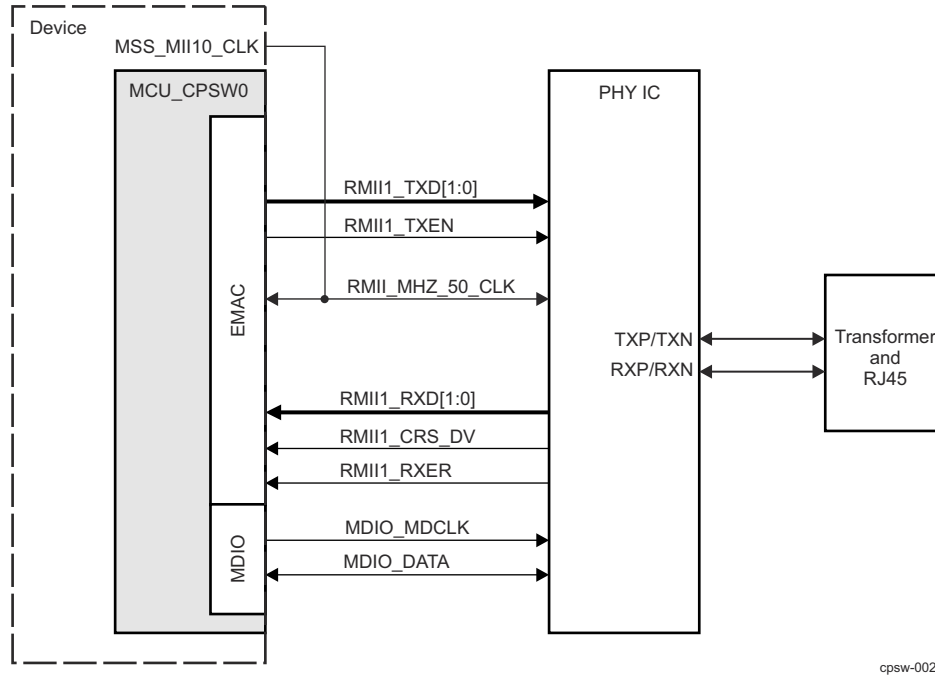
<b>AVB</b>	Audio Video Bridging
<b>AVBTP</b>	Audio Video Bridging Transport Protocol
<b>BMCA</b>	Best Master Clock Algorithm
<b>CFI</b>	Canonical Format Indicator
<b>CPPI</b>	Communications Port Programming Interface
<b>DLR</b>	Device Level Ring
<b>DSCP</b>	Differentiated Services Code Point
<b>EEE</b>	Energy Efficient Ethernet
<b>EMAC</b>	Ethernet Media Access Control
<b>EOP</b>	End of Packet
<b>EOQ</b>	End of Queue
<b>IPG</b>	Inter-Packet Gap
<b>LPI</b>	Low Power Indicator
<b>MDIO</b>	Management Data Input/Output
<b>MOF</b>	Middle of Frame
<b>OUI</b>	Organizationally Unique Identifier
<b>PTP</b>	Precision Time Protocol
<b>RMON</b>	Remote Monitoring
<b>RTCP</b>	RTP Control Protocol
<b>RTP</b>	Real-time Transport Protocol
<b>SCR</b>	Switched Central Resource
<b>SRP</b>	Stream Reservation Protocol
<b>TOS</b>	Type of Service
<b>VLAN</b>	Virtual Local Area Network
<b>CPSW_2G</b>	CPSW two port

## 9.2 MCU\_CPSW0 Environment

### 9.2.1 MCU\_CPSW0 RMII Interface

Figure 9-2 shows a device with integrated RMII and MDIO interface connection in a typical system. The individual MCU\_CPSW0 and MDIO signals for the RMII interface are summarized in Table 9-1.

For more information, refer to either the IEEE 802.3 standard or ISO/IEC 8802-3:2000(E).



**Figure 9-2. RMII Interface Typical Application**

**Table 9-1. RMII I/O Description**

Signal	Device Pin	I/O <sup>(1)</sup>	Description
RMII_TXD[1:0]	MCU_RMII_TXD[1:0]	O	Transmit data. The transmit data pins are a collection of 2 bits of data. TXD0 is the least-significant bit (LSB). The signals are synchronized by RMII_MHZ_50_CLK and valid only when RMII_TXEN is asserted.
RMII_TXEN	MCU_RMII_TX_EN	O	RMII transmit enable. The transmit enable signal indicates that the MCU_RMII_TXD pins are generating data for use by the PHY. RMII_TXEN is synchronous to RMII_MHZ_50_CLK.
RMII_MHZ_50_CLK	MCU_RMII_REF_CLK	I	RMII 50MHz reference clock.  The reference clock is used to synchronize all RMII signals. RMII_MHZ_50_CLK must be continuous and fixed at 50 MHz. This bit controls the clock source MSS_CTRL::CPSW_CONTROL::CPSW_CONTROL_RMII_REF_CLK_OE_N
RMII_RXD[1:0]	MCU_RMII_RXD[1:0]	I	Receive data. The receive data pins are a collection of 2 bits of data. RXD0 is the least-significant bit (LSB). The signals are synchronized by RMII_MHZ_50_CLK and valid only when RMII_CRS_DV is asserted and RMII_RXER is de-asserted.
RMII_CRS_DV	MCU_RMII_CRS_DV	I	Carrier sense/receive data valid. Multiplexed signal between carrier sense and receive data valid.
RMII_RXER	MCU_RMII_RX_ER	I	Receive error. The receive error signal is asserted to indicate that an error was detected in the received frame.
MDIO_MDCLK	MCU_MDIO_CLK	O	Management data clock (MDIO_MDCLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MCU_MDIO0_DATA pin.

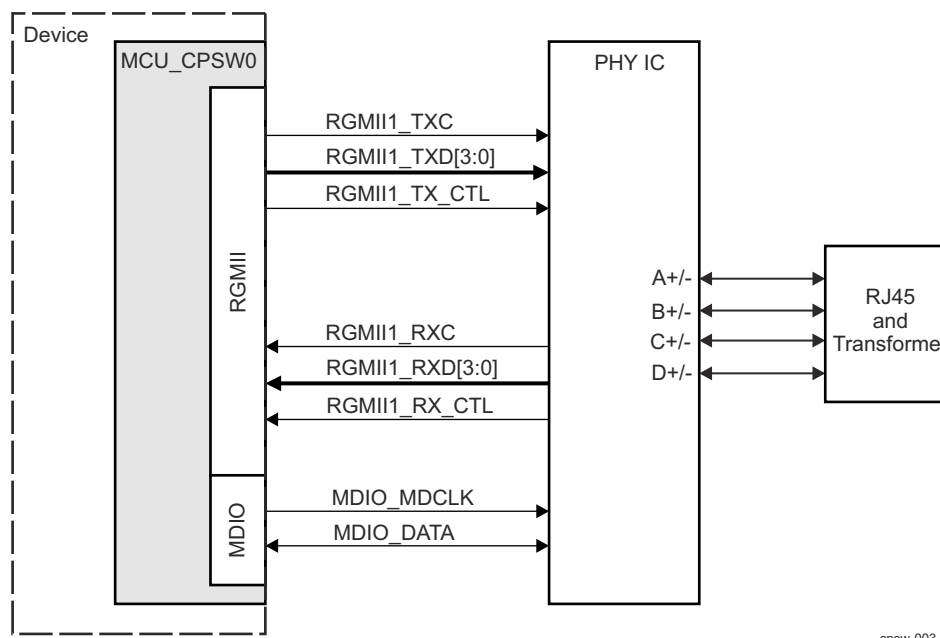
**Table 9-1. RMII I/O Description (continued)**

Signal	Device Pin	I/O <sup>(1)</sup>	Description
MDIO_DATA	MCU_MDIO_DATA	I/O	MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MCU_MDIO0_DATA pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

(1) I = Input; O = Output

### 9.2.2 MCU\_CPSW0 RGMII Interface

Figure 9-3 shows a device with integrated RGMII and MDIO interface connection in a typical system. The individual MCU\_CPSW0 and MDIO signals for the RGMII interface are summarized in Table 9-2.



cpsw-003

**Figure 9-3. RGMII Interface Typical Application**
**Table 9-2. RGMII I/O Description**

Signal	Device Pin(s)	I/O <sup>(1)</sup>	Description
RGMII_TXD[3:0]	MCU_RGMII_TD[3:0]	O	The transmit data pins are a collection of 4 bits of data. TD0 is the least-significant bit (LSB). The signals are valid only when RGMII_TX_CTL is asserted.
RGMII_TX_CTL	MCU_RGMII_TX_CTL	O	Transmit Control/enable. The transmit enable signal indicates that the TD pins are generating data for use by the PHY.
RGMII_TXC	MCU_RGMII_TXC	O	The transmit reference clock. The clock is 2.5 MHz at 10 Mbps operation, 25 MHz at 100 Mbps operation, and 125 MHz at 1000 Mbps* of operation.
RGMII_RXD[3:0]	MCU_RGMII_RD[3:0]	I	The receive data pins are a collection of 4 bits of data. RD0 is the least-significant bit (LSB). The signals are valid only when RGMII_RX_CTL is asserted
RGMII_RX_CTL	MCU_RGMII_RX_CTL	I	The receive data valid/control signal indicates that the RD pins are nibble data for use by the EMAC.
RGMII_RXC	MCU_RGMII_RXC	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation, 25 MHz at 100 Mbps operation, 125 MHz at 1000 Mbps* of operation.

**Table 9-2. RGMII I/O Description (continued)**

Signal	Device Pin(s)	I/O <sup>(1)</sup>	Description
MDIO_MDCLK	MCU_MDIO_CLK	O	Management data clock (MDIO_MDCLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MCU_MDIO0_DATA pin.
MDIO_DATA	MCU_MDIO_DATA	I/O	The MCU_MDIO0_DATA pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MCU_MDIO0_DATA pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

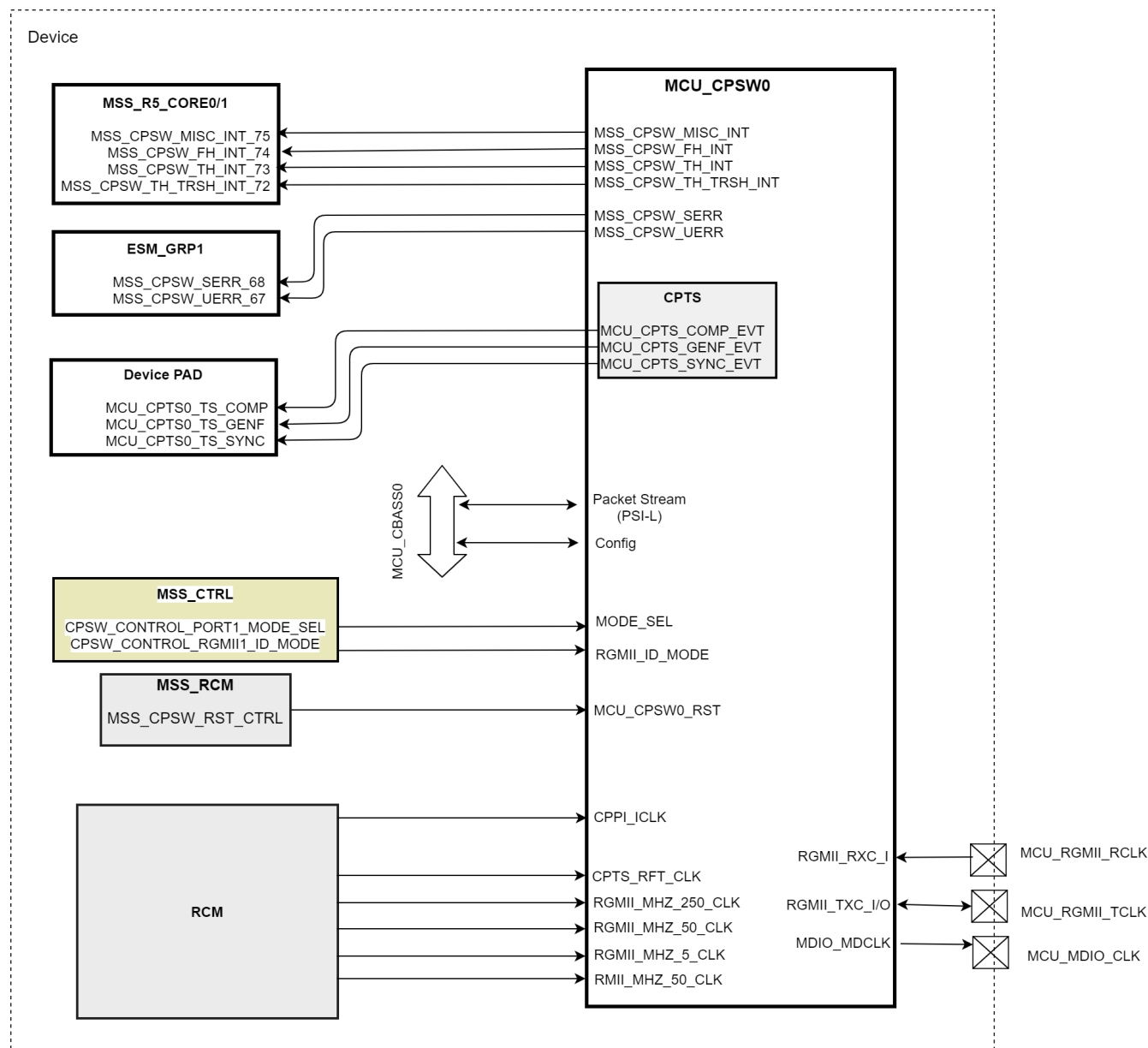
(1) I = Input; O = Output

#### Note

The Control Module registers assign the specific function to the device pads. For more information on Control Module settings, see , Pad Configuration Registers in *Control Module (CTRL\_MMR)* and the device-specific Datasheet.

## 9.3 MCU\_CPSW0 Integration

Figure 9-4 shows the integration of the MCU\_CPSW0 module in the device.



**Figure 9-4. MCU\_CPSW0 Integration**

Below Tables summarize the integration of the MCU\_CPSW0 module in the device.

**Table 9-3. MCU\_CPSW0 Clocks and Resets**

Clocks			
Module Instance	Module Clock Input	Reference/Source	Description
MCU_CPSW0	CPPI_CLK	MSS_RCM::CPSW_CLK	CPPI packet streaming interface clock. Main clock for MCU_CPSW0.
	CPTS_RFT_CLK	MSS_RCM::MSS_CPTS_CLK	Time sync Reference Clock 400/250/divided
	RGMII_MHZ_5_CLK	MSS_RCM::MSS_MII10_CLK	5-MHz RGMII reference clock.
	RGMII_MHZ_50_CLK	MSS_RCM::MSS_MII100_CLK	50-MHz RGMII reference clock.

**Table 9-3. MCU\_CPSW0 Clocks and Resets (continued)**

RGMII_MHZ_250_CLK	MSS_RCM::MSS_RGMII_CLK	250-MHz RGMII reference clock.
RMII_MHZ_50_CLK	MSS_RCM::MSS_MII100_CLK	50-MHz RMII reference clock.

Resets			
Module Instance	Module Reset Input	Reference/Source	Description
MCU_CPSW0	MCU_CPSW0_RST	MCSS_RCM::MSS_CPSW_RST_CTRL	IP Reset

**Table 9-4. MCU\_CPSW0 Hardware Requests**

Interrupt Requests				
Module Instance	Module Interrupt Signal	Destination Interrupt Input	Description	Type
MCU_CPSW0	MCU_CPSW_STAT_PEND	MSS_CPSW_MISC_INT_75	MCU_CPSW0 statistic pending interrupt 0	Pulse
	MCU_CPSW_EVNT_PEND	MSS_CPSW_MISC_INT_75	MCU_CPSW0 event pending interrupt	Pulse
	MCU_CPSW_MDIO_INTR	MSS_CPSW_MISC_INT_75	MCU_CPSW0 MDIO interrupt	Pulse
	MCU_CPSW_THost_INT	MSS_CPSW_TH_INT_73	MCU_CPSW0 T-host interrupt	Pulse
	MCU_CPSW_FHost_INT	MSS_CPSW_FH_INT_74	MCU_CPSW0 F-host interrupt	Pulse
	MCU_CPSW_TH_TRSH_INT	MSS_CPSW_TH_TRSH_INT_72	MCU_CPSW0 TH-threshold interrupt	Pulse
	MCU_CPSW_ECC_SEC_INT	MCU_ESM_EVT_68	MCU_CPSW0 SEC ECC error interrupt	Pulse
	MCU_CPSW_ECC_DED_INT	MCU_ESM_EVT_67	MCU_CPSW0 SEC DED error interrupt	Pulse

Time Sync and Compare Events				
Module Instance	Module Event	Device Pin	Description	Type
MCU_CPSW0	MCU_CPSW_COMP_EVT	MCU_CPTS0_TS_COMP	MCU_CPSW0 compare event interrupt	Edge
	MCU_CPSW_GENF_EVT	MCU_CPTS0_TS_GENF	MCU_CPSW0 CPTS generator function event interrupt	Edge
	MCU_CPSW_SYNC_EVT	MCU_CPTS0_TS_SYNC	MCU_CPSW0 CPTS sync event interrupt	Edge

### Note

For more information about interrupts, see *Local Interrupt Controller*.

For more information on the interconnects, see [Chapter 3](#).

For more information on the power, reset and clock management, see the corresponding sections within [Chapter 5](#).

For more information on the device interrupt controllers, see *Interrupt Controllers*.

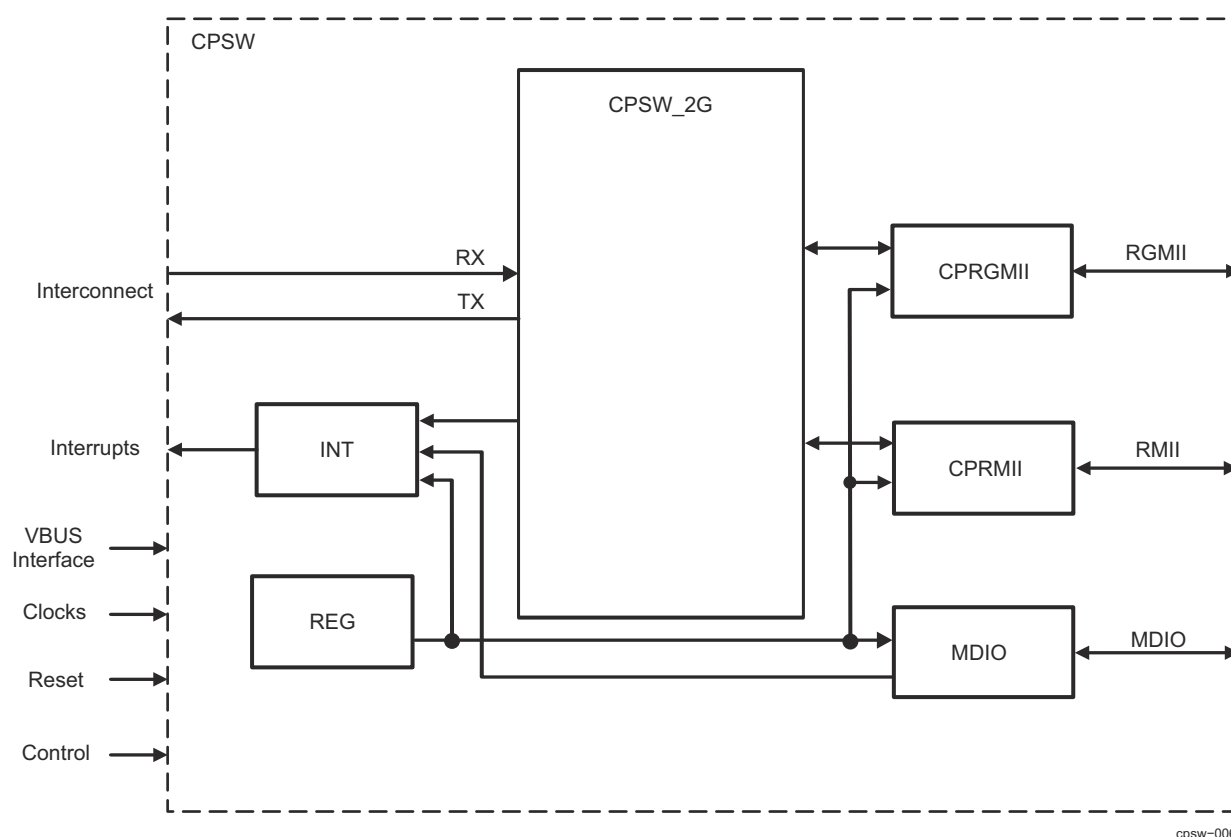
## 9.4 MCU\_CPSW0 Functional Description

The two-port switch Ethernet subsystem modules (CPSW) are compliant to the IEEE Std 802.3 Specification. CPSW top level functional block diagram is shown in [Figure 9-5](#).

### 9.4.1 Functional Block Diagram

The two-port Ethernet subsystem consists of:

- CPSW\_2G
- One RGMII interface module
- One RMII interface module
- One Host Port 0 CPPI Packet Streaming Interface
- CPSW subsystem control registers (REG)
- One MDIO interface module
- One Interrupt Controller module



**Figure 9-5. CPSW Top Level Block Diagram**

### 9.4.2 CPSW Ports

The Ethernet Subsystem has two ports. Port 0 is the Host port (internal to the Subsystem). Port 1 is the external port connected to RGMII, or RMII interfaces as per the interface selected.

Naming conventions followed in this chapter:

- Port0 is referred to the CPPI CPDMA Host Port
- Port1 is referred to the interfaces RGMII/RMII

#### 9.4.2.1 Ethernet Port Operation



#### 9.4.2.1.1 Interface Mode Selection

The two-port switch (CPSW) Ethernet Subsystem has one 10/100/1000 Ethernet port with selectable MII, RMII, and RGMII interfaces. These interfaces can be enabled by

**MSS\_CTRL:CPSW\_CONTROL:CPSW\_CONTROL\_PORT1\_MODE\_SEL**

For specific interface mode further features is selected by configuring the Ethernet bits (EXT\_EN, GIG, FULLDUPLEX, IFCTL\_A) in the **PN\_MAC\_CONTROL** register.

See the device-specific Datasheet for configuring the pin mux mode as per the interface selected.

CPSW_CONTROL_PORT1_MODE_SEL	Link Type
00	GMII/MII
01	RMII
10	RGMII

#### 9.4.2.1.2 RMII Operation

The **IFCTL\_A** bit determines the RMII link speed (0=10mbps, 1=100mbps). The **FULLDUPLEX** bit controls RMII duplexity. The **IFCTL\_B** is not used.

#### 9.4.2.1.3 RGMII Operation

Each port RGMII interface can operate in a forced mode or an in-band mode as determined by the **EXT\_EN** bit. Forced mode is with **EXT\_EN** cleared. In-band mode is with **EXT\_EN** set.

##### 9.4.2.1.3.1 RGMII In-Band Operation

A port CPRGMII is operating in the in-band mode of operation when the **EXT\_EN** is set. The link status, duplexity, and speed are determined from the RGMII input data stream as defined in the RGMII specification and can be read in the **SS\_RGMII1\_STATUS** register. The link speed is indicated as shown in [Table 9-5](#).

**Table 9-5. RGMII Link Speed**

SPEED(1:0)	Link Speed
00	10 Mbs mode
01	100 Mbs mode
10	1000 Mbs mode
11	reserved

##### 9.4.2.1.3.2 RGMII Forced Mode Operation

#### 9.4.3 Clocking

##### 9.4.3.1 Subsystem Clocking

CPSW clocking summary is shown in [Section 9.3, CPSW Integration](#).

##### 9.4.3.2 Interface Clocking

Data is transmitted and received with respect to the reference clocks of the interface pins.

##### 9.4.3.2.1 RGMII Interface Clocking

RGMII\_RXC, RGMII\_TXC frequencies are:

- 2.5 MHz at 10 Mbps
- 25 MHz at 100 Mbps

##### 9.4.3.2.2 RMII Interface Clocking

RMII interface clock RMII\_50MHZ\_CLK frequency is:

- 50 MHz at 10 Mbps

- 50 MHz at 100 Mbps

CTRLMMR\_MCU\_CLKOUT0\_CTRL[4]CLK\_EN and CTRLMMR\_MCU\_CLKOUT0\_CTRL[0]CLK\_SEL bits are used to enable and select the clock source for MCU\_CLKOUT pin.

#### 9.4.3.2.3 MDIO Clocking

The MDIO clock is based on a divide-down of the interface (CPPI\_ICLK) clock. The application software or driver must control the divide-down value.

See the CPSW\_MDIO\_CONTROL\_REG register for configuring the Clock Divider ([15-0]CLKDIV) value.

### 9.4.4 Interrupt Functionality

**Table 9-6. Interrupt Functionality**

Interrupt	Description
MSS_CPSW_FH_INT	FHost (from host to Ethernet) interrupt
MSS_CPSW_TH_INT	THost (from Ethernet to host) interrupt
MSS_CPSW_TH_TRSH_INT	THost (from Ethernet to host) Threshold interrupt
MSS_CPSW_MISC_INT	Miscellaneous interrupt
MSS_CPSW_SERR	ECC SEC interrupt – output from CPSW ECC module. This interrupt is also included in the <b>MSS_CPSW_MISC_INT</b> if enabled or this interrupt can be used separately.
MSS_CPSW_UERR	ECC DED interrupt – output from CPSW ECC module. This interrupt is also included in the <b>MSS_CPSW_MISC_INT</b> if enabled or this interrupt can be used separately.

#### 9.4.4.1 MSS\_CPSW\_TH\_TRSH\_INT Interrupt Description

The **MSS\_CPSW\_TH\_TRSH\_INT** interrupts are each an immediate (non-paced) pulse interrupt selected from the CPSW\_2G **TH\_THRESH\_PEND[7:0]** interrupts. The THost threshold pending interrupt(s) is selected by setting one or more bits in the **SS\_TH\_THRESH\_PULSE\_EN\_REG[7:0]** register. The masked interrupt status can be read in the **SS\_TH\_THRESH\_PULSE\_STATUS\_REG[7:0]** register. Upon reception of an interrupt, software should perform the following:

- Read the **SS\_TH\_THRESH\_PULSE\_STATUS\_REG[7:0]** register to determine which channel(s) caused the interrupt.
- Process THost packets in order to add more buffers to any channel that is below the threshold value.
- Write the CPSW\_2G completion pointer(s) to acknowledge the CPDMA interrupt.
- Write 0x0 to the CPDMA\_EOI\_VECTOR register in the CPSW\_2G slave address space to acknowledge the subsystem interrupt.

#### 9.4.4.2 MSS\_CPSW\_TH\_INT Interrupt Description

The **MSS\_CPSW\_TH\_INT** interrupts are each a paced pulse interrupt selected from the CPSW\_2G **TH\_PEND[7:0]** interrupts. The THost pending interrupt(s) is selected by setting one or more bits in the **SS\_TH\_PULSE\_EN\_REG[7:0]** register. The masked interrupt status can be read in the **SS\_TH\_PULSE\_STATUS\_REG[7:0]** register. Upon reception of an interrupt, software should perform the following:

- Read the **SS\_TH\_PULSE\_STATUS\_REG[7:0]** register to determine which channel(s) caused the interrupt.
- Process THost packets for the interrupting channel(s) to acknowledge the CPDMA interrupt.
- Write the CPSW\_2G completion pointer(s).
- Write 0x1 to the CPDMA\_EOI\_VECTOR\_REG register in the CPSW\_2G slave address space to acknowledge the subsystem interrupt.

#### 9.4.4.3 MSS\_CPSW\_FH\_INT Interrupt Description

The **MSS\_CPSW\_FH\_INT** interrupts are each a pulse interrupt selected from the CPSW\_2G **FH\_PEND[7:0]** interrupts. The transmit pending interrupt(s) is selected by setting one or more bits in the **SS\_FH\_PULSE\_EN\_REG[7:0]** register. The masked interrupt status can be read in the

**SS\_FH\_PULSE\_STATUS\_REG[7:0]** register. Upon reception of an interrupt, software should perform the following:

- Read the **SS\_FH\_PULSE\_STATUS\_REG[7:0]** register to determine which channel(s) caused the interrupt.
- Process THost packets for the interrupting channel(s).
- Write the CPSW\_2G completion pointer(s) to acknowledge the CPDMA interrupt.
- Write 0x2 to the **CPDMA\_EOI\_VECTOR\_REG** register in the CPSW\_2G slave address space to acknowledge the subsystem interrupt.

#### 9.4.4.4 MSS\_CPSW\_MISC\_INT Interrupt Description

The **MSS\_CPSW\_MISC\_INT** interrupts are each an immediate (non-paced) pulse interrupt selected from the miscellaneous interrupts (**DED\_PEND**, **SEC\_PEND**, **EVNT\_PEND**, **STAT\_PEND**, **HOST\_PEND**, **MDIO\_LINKINT**, **MDIO\_USERINT**). The miscellaneous interrupt bits are enabled by setting one or more bits in the **SS\_MISC\_EN\_REG** register. The masked interrupt status can be read in the **SS\_MISC\_STATUS\_REG** register. Upon reception of an interrupt, software should perform the following:

- Read the **SS\_MISC\_STATUS\_REG** register to determine the cause of the interrupt.
  - **MDIO\_USERINT** is the logical “OR” of the CPSW\_2G MDIO **MDIO\_USERINT[1:0]**.
  - **MDIO\_LINKINT** is the logical “OR” of the CPSW\_2G MDIO **MDIO\_LINKINT[1:0]**
  - **HOST\_PEND** is the CPSW\_2G CPDMA host error interrupt.
  - **STAT\_PEND** is the logical “OR” of the CPSW\_2G statistics interrupt **STAT\_PEND[2:0]**.
  - **EVNT\_PEND** is the CPSW\_2G CPTS event interrupt.
  - **SEC\_PEND** is the ECC Single bit error interrupt.
  - **DED\_PEND** is the ECC double bit error interrupt.
- Process the interrupt.
- Write 0x3 to the **CPDMA\_EOI\_VECTOR** register in the CPSW\_2G slave address space to acknowledge the subsystem interrupt.

#### 9.4.4.5 ECC SEC ESM Interrupt (MSS\_CPSW\_SERR)

ESM interrupt indicating a CPSW\_2G ECC single error has been detected and corrected. Although the **MSS\_CPSW\_SERR** value can be read in the **CPDMA\_IN\_VECTOR** register, there is no EOI associated with this interrupt. It is a direct output of the CPSW\_2G ECC aggregator module. MSS can enable this interrupt from **SS\_MISC\_EN\_REG[5]** and read status from **SS\_TH\_PULSE\_STATUS\_REG[5]**

#### 9.4.4.6 ECC DED ESM Interrupt (MSS\_CPSW\_UERR)

ESM interrupt indicating a CPSW\_2G ECC double error has been detected. Although the **MSS\_CPSW\_UERR** value can be read in the **CPDMA\_IN\_VECTOR** register, there is no EOI associated with this interrupt. It is a direct output of the CPSW\_2G ECC aggregator module. MSS can enable this interrupt from **SS\_MISC\_EN\_REG[6]** and read status from **SS\_MISC\_STATUS\_REG[6]**.

#### 9.4.4.7 THost Threshold Interrupts for CPDMA

Each of the eight THost channels has a corresponding THost threshold interrupt (**TH\_THRESH\_PEND[7:0]**). The THost threshold interrupts are level interrupts that remain asserted until the triggering condition is cleared by the host. Each of the eight threshold interrupts may be individually enabled by setting to one the appropriate bit in the **CPDMA\_TH\_INTSTAT\_SET** register. Each of the eight channel interrupts may be individually disabled by clearing to zero the appropriate bit in the **CPDMA\_TH\_INTSTAT\_CLR** register. The raw and masked interrupt receive interrupt status may be read by reading the **CPDMA\_TH\_INTSTAT\_RAW** and **CPDMA\_TH\_INTSTAT\_MASKED** registers respectively. A **TH\_THRESH\_PEND[7:0]** interrupt bit is asserted when enabled and when the channel's associated freebuffer count (**CPDMA\_TH(0/7)\_FREEBUFFER**) is less than or equal to the channel's associated flow control threshold register (**CPDMA\_TH(0/7)\_PENDTHRESH**). The threshold interrupts are intended to give the host an indication that resources are running low for a particular channel(s).

#### 9.4.4.8 Thost Packet Completion Interrupts for CPDMA

The THost DMA controller has eight channels with each channel having a corresponding interrupt (**TH\_PEND[7:0]**). The THost interrupts are level interrupts that remain asserted until cleared by the host.

Each of the eight channel interrupts may be individually enabled by setting to one the appropriate bit in the **CPDMA\_TH\_INTSTAT\_SET** register. Each of the eight channel interrupts may be individually disabled by clearing to zero the appropriate bit in the **CPDMA\_TH\_INTSTAT\_CLR** register. The raw and masked interrupt interrupt status may be read by reading the **CPDMA\_TH\_INTSTAT\_RAW** and **CPDMA\_TH\_INTSTAT\_MASKED** registers respectively. When a packet transfer is complete, the CPDMA issues an interrupt to the host by writing the packet's last buffer descriptor address to the appropriate channel queue's **CPDMA\_TH(0..7)\_CP** completion pointer. The interrupt is generated by the write, regardless of the value written. Upon interrupt reception, the host processes one or more packets from the queue and then acknowledges one or more interrupt(s) by writing the address of the last buffer descriptor processed to the queue's associated **CPDMA\_TH(0..7)\_CP** Completion Pointer. If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted which means that the CPDMA has transferred more packets than the host has processed interrupts for. If the host written buffer address value is equal to the port written value then the host has processed all packets that the CPDMA has transferred and the level interrupt is deasserted. The CPDMA write to the completion pointer actually stores the value. The host written value is actually not written to the register location. The host written value is compared to the register contents which was written by the CPDMA. If the two values are equal then the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

#### 9.4.4.9 FHost Packet Completion Interrupts for CPDMA

The FHost DMA controller has eight channels with each channel having a corresponding interrupt (**FH\_PEND[7:0]**). The FHost interrupts are level interrupts that remain asserted until cleared by the host. Each of the eight channel interrupts may be individually enabled by setting to one the appropriate bit in the **CPDMA\_FH\_INTSTAT\_MASKED\_SET** register. Each of the eight channel interrupts may be individually disabled by clearing to zero the appropriate bit in the **CPDMA\_FH\_INTSTAT\_MASKED\_CLR** register. The raw and masked interrupt status may be read by reading the **CPDMA\_FH\_INTSTAT\_RAW** and **CPDMA\_FH\_INTSTAT\_MASKED** registers respectively. When each packet transfer is complete, the CPDMA issues an interrupt to the host by writing the packet's last buffer descriptor address to the appropriate channel **CPDMA\_FH(0..7)\_CP** completion pointer register. The interrupt is generated by the write, regardless of the value written. Upon interrupt reception, the host processes one or more packets from the queue and then acknowledges an interrupt by writing the address of the last buffer descriptor processed to the queue's associated FHost Completion Pointer. If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted which means that the CPDMA has transferred more packets than the host has processed interrupts for. If the host written buffer address value is equal to the port written value then the host has processed all packets that the CPDMA has transferred and the level interrupt is deasserted. The CPDMA write to the completion pointer actually stores the value. The host written value is actually not written to the register location. The host written value is compared to the register contents which was written by the CPDMA and if the two values are equal then the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

#### 9.4.5 Media Independent Interfaces (XGMII/GMII)

For the purposes of this document, Ethernet transmit is egress and Ethernet receive is ingress.

##### 9.4.5.1 Receive Control

Data received from the PHY is interpreted and forwarded by the MAC into the switch. Interpretation involves detection and removal of the preamble and start of frame delimiter, extraction of the address and frame length, data handling, error checking and reporting, cyclic redundancy checking (CRC), and statistics control signal generation.

##### 9.4.5.1.1 Receive Inter-Frame Interval

The 802.3 required inter-packet gap (IPG) is 24 GMII clocks (96 bit times) for 10/100 Mbit modes, and 12 GMII clocks (96 bit times) for 1000 Mbit mode. However, the MAC can tolerate a reduced IPG (2 GMII clocks in 10/100 mode and 5 GMII clocks in 1000 mode) with a correct preamble and start frame delimiter.

This interval between frames must comprise (in the following order):

1. An Inter-Packet Gap (IPG).
2. A seven octet preamble (all octets 0x55).
3. A one octet start frame delimiter (0x5d).

#### 9.4.5.1.2 Received Frame Classification

Received frames are proper (good) frames if they are between 64 and **PN\_RX\_MAXLEN** in length (inclusive) and contain no errors (code/align/CRC).

Received frames are long frames if their frame count exceeds the value in the **PN\_RX\_MAXLEN** register. The **PN\_RX\_MAXLEN** register reset (default) value is 1518 (dec). Long received frames are either oversized or jabber frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment errors are jabber frames.

Received frames are short frames if their frame count is less than 64 bytes. Short frames that contain no errors are runt frames. Short frames with CRC, code, or alignment errors are fragment frames. If **RX\_CSF\_EN** in **PN\_MAC\_CONTROL** is set, undersized frames from 33 to 63 bytes will be forwarded only to the host on a best effort basis (meaning that the ALE may or may not be able to keep up with the packet rate and the short packet may be dropped due to bandwidth limitations). If **RX\_CSF\_EN** and **RX\_CEF\_EN** in **PN\_MAC\_CONTROL** are set, fragment frames from 33 to 63 bytes will also be forwarded only to the host on a best effort basis. Ethernet port received frames shorter than 33 bytes are dropped in all cases.

A received long packet will always contain **PN\_RX\_MAXLEN** number of bytes transferred to memory (if **RX\_CEF\_EN** = 1). Examples with **PN\_RX\_MAXLEN** = 1518 is below:

- If the frame length is 1518, then the packet is not a long packet and there will be 1518 bytes transferred to memory.
- If the frame length is 1519, there will be 1518 bytes transferred to memory. The last three bytes will be the first three CRC bytes.
- If the frame length is 1520, there will be 1518 bytes transferred to memory. The last two bytes will be the first two CRC bytes.
- If the frame length is 1521, there will be 1518 bytes transferred to memory. The last byte will be the first CRC byte.
- If the frame length is 1522, there will be 1518 bytes transferred to memory. The last byte will be the last data byte.

#### 9.4.5.2 Transmit Control

A jam sequence is output if a collision is detected on a transmit packet in half-duplex mode. If the collision was late (after the first 64 bytes have been transmitted) then the collision is ignored. If the collision is not late, then the controller will back off before retrying the frame transmission. When operating in full duplex mode the carrier sense (CRS) and collision sensing modes are disabled.

##### 9.4.5.2.1 Adaptive Performance Optimization (APO)

The Ethernet MAC port incorporates Adaptive Performance Optimization (APO) logic that may be enabled by setting the **TX\_PACE** bit in the **PN\_MAC\_CONTROL** register. Transmission pacing to enhance performance is enabled when set. Adaptive performance pacing introduces delays into the normal transmission of frames, delaying transmission attempts between stations, reducing the probability of collisions occurring during heavy traffic (as indicated by frame deferrals and collisions) thereby increasing the chance of successful transmission.

When a frame is deferred, suffers a single collision, multiple collisions or excessive collisions, the pacing counter is loaded with an initial value of 31. When a frame is transmitted successfully (without experiencing a deferral, single collision, multiple collision or excessive collision) the pacing counter is decremented by one, down to zero.

With pacing enabled, a new frame is permitted to immediately (after one IPG) attempt transmission only if the pacing counter is zero. If the pacing counter is non-zero, the frame is delayed by the pacing delay, a delay of



approximately four inter-packet gap delays. APO only affects the IPG preceding the first attempt at transmitting a frame. It does not affect the back-off algorithm for retransmitted frames.

#### 9.4.5.2.2 Inter-Packet-Gap Enforcement

The measurement reference for the IPG of 96 bit times is changed depending on frame traffic conditions. If a frame is successfully transmitted without collision, and **MCRS** is de-asserted within approximately 48 bit times of **MTXEN** being de-asserted, then 96 bit times is measured from **MTXEN**. If the frame suffered a collision, or if **MCRS** is not de-asserted until more than approximately 48 bit times after **MTXEN** is de-asserted, then 96 bit times (approximately, but not less) is measured from **MCRS**.

The Ethernet port transmit inter-packet gap (IPG) may be shortened by eight bit times when short gap is enabled and triggered. Setting the **TX\_SHORT\_GAP\_ENABLE** bit each **PN\_MAC\_CONTROL** register enables the gap to be shortened when triggered. The condition is triggered when the ports associated transmit packet FIFO has a user defined number of FIFO blocks used. The associated transmit FIFO blocks used value determines if the gap is shortened, and so on. The **GAP\_THRESH** register value determines the short gap threshold. If the FIFO blocks used is greater than or equal to the **GAP\_THRESH** value then short gap is triggered.

#### 9.4.5.2.3 Programmable Transmit Inter-Packet Gap

The transmit inter-packet gap (IPG) is programmable through the **PN\_MAC\_TX\_GAP** register. The default value is decimal 12. The transmit IPG may be increased to the maximum value of 0x1ff. Increasing the IPG is not compatible with transmit pacing. The short gap feature will override the increased gap value, so the short gap feature may not be compatible with an increased IPG.

#### 9.4.5.2.4 Back Off

The Ethernet Mac implements the 802.3 binary exponential back-off algorithm for half-duplex based collisions.

#### 9.4.5.3 Emulation Control

The emulation control input (**TBEMUSUP**) and register bits (**SOFT** and **FREE** in the **PN\_MAC\_EMCONTROL** register) allow Mac operation to be suspended. When the emulation suspend state is entered, the MAC will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For receive, frames that are detected by the MAC after the suspend state is entered are ignored. Emulation control is implemented for compatibility with other peripherals. [Table 9-7](#) shows the operations of emulation control input and register bits.

**Table 9-7. Emulation Control Input and Register Bits**

TBEMUSUP	SOFT	FREE	Description
0	X	X	Normal Operation
1	0	0	Normal Operation
1	1	0	Emulation Suspend
1	X	1	Normal Operation

#### 9.4.5.4 Command IDLE

The **CMD\_IDLE** bit in the **PN\_MAC\_CONTROL** register allows MAC operation to be suspended by software. When the idle state is commanded, the MAC will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. Received frames that are detected after the suspend state is entered are ignored (dropped without any processing or statistics). Commanded idle is similar in operation to emulation control and hardware clock stop.

#### 9.4.6 Software IDLE

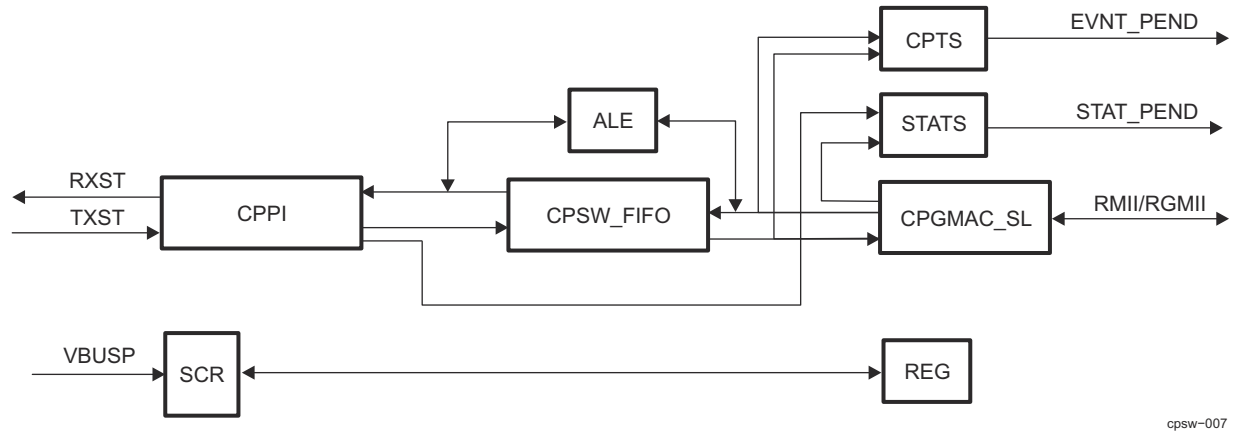
The submodule software idle register bits enable CPSW operation to be completely or partially suspended by software control. There are two CPSW submodules that contain software idle register bits. Each of the two submodules may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled submodule until the idle command is removed. The CPSW module enters the idle state when all two submodules are commanded to enter and have

entered the idle state. Idle status is determined by reading or polling the two submodule idle bits. The CPSW\_2G is in the idle state when all two submodules are in the idle state. The CPSW\_SOFT\_IDLE\_REG[0] SOFT\_IDLE bit may be set if desired after the submodules are in the idle state. The SOFT\_IDLE bit causes packets to not be transferred from one FIFO to another FIFO internal to the switch.

#### 9.4.7 CPSW\_2G

The CPSW\_2G RMII/ RGMII interface is compliant to the IEEE Std 802.3 Specification.

The CPSW\_2G contains one Ethernet port interface (Ethernet port 1), one CPPI packet streaming interface host port (port 0), Common Platform Time Sync (CPTS), ALE Engine and Statistics (STATS). A top-level block diagram of the CPSW\_2G is shown in [Figure 9-6](#).



**Figure 9-6. CPSW\_2G Block Diagram**

##### 9.4.7.1 Address Lookup Engine (ALE)

The Address Lookup Engine (ALE) is a sub-block of the CPSW Switch and it processes all received packets and determines to which port(s) the packet should be forwarded. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded. The ALE outputs the port mask to the switch fabric that indicates the port(s) the packet should be forwarded to. The ALE is enabled when the ENABLE bit in the ALE\_CONTROL\_REG register is set. All packets are dropped when the ENABLE bit is cleared to 0.

###### 9.4.7.1.1 Error Handling

In normal operation, the Ethernet port is configured to issue an abort, instead of an end of packet, at the end of a packet that contains an error (runt, frag, oversize, jabber, crc, alignment, code etc.) or at the end of a MAC control packet. However, when the PN\_MAC\_CONTROL\_REG configuration bit(s) RX\_CEF\_EN, RX\_CSF\_EN, or RX\_CMF\_EN are set, error frames, short frames or MAC control frames have a normal end of packet instead of an abort at the end of the packet. When the ALE receives a packet that contains errors (due to a set header error bit), or a MAC control frame and does not receive an abort, the packet will be forwarded only to the host port (port 0). Packets with errors that are forwarded to the host have no VLAN untagging or drop due to rate limiting. No ALE learning occurs on packets with errors or mac control frames. Learning is based on source address and lookup is based on destination address. Directed packets from the host are not learned, updated, or touched.

###### 9.4.7.1.2 Bypass Operations

The ALE may be configured to operate in bypass mode by setting the BYPASS bit in the ALE\_CONTROL\_REG register. When in bypass mode, all Ethernet port received packets are forwarded only to the host port (port 0). In bypass mode, the ALE processes host port transmit packets the same as in normal mode. In general, packets would be directed by the host in bypass mode.

### 9.4.7.1.3 OUI Deny or Accept

The ALE may be configured to operate in OUI deny mode by setting the enable\_oui\_deny bit in the ALE\_CONTROL register. When in OUI deny mode, any packet with a non-matching OUI source address will be dropped to the host unless the packet destination address matches with a supervisory table entry. (Non-matching OUI source address broadcast/multicast packets will be dropped to the host unless the packet destination address is entered into the table with the super bit set. Non-matching OUI source address unicast packets will be dropped to the host unless the unicast destination address is in the table with block and Secure both set). When enable\_oui\_deny is cleared, any packet source address matching an OUI address table entry will be dropped to the host unless the destination address matches with a supervisory address table entry. (Broadcast packets matching the OUI source address will be dropped to the host unless the broadcast destination address is entered into the table with the super bit set. Unicast packets matching the OUI source address will be dropped to the host unless the unicast destination address is in the table with block and Secure both set)

### 9.4.7.1.4 Statistics Counting

ALE sends many statistics along with the frame routing so the CPSW can count them on a per port basis. The events specified through the CPSW\_ALE\_STAT\_DIAG[3-0] STAT\_DIAG field are individually counted in CPSW per port statistics counters.

### 9.4.7.1.5 Supervisory packets

Multicast supervisory packets are designated by the SUPER bit in the table entry. Unicast supervisory packets are indicated when BLOCK and SECURE are both set. Supervisory packets are not dropped due to rate limiting, OUI, or VLAN processing. The purpose of supervisory packets is to allow packets that would be otherwise blocked to be forwarded for special purposes.

### 9.4.7.1.6 ALE Lookup Table Entry Definitions

The ALE table contains multiple table entry types. Each table entry represents a free entry, an address, a VLAN, an address/VLAN pair, or an OUI address. Software should ensure that there are not double address entries in the table. The double entry used would be indeterminate. Reserved table bits must be written with zeroes.

Source Address learning occurs for packets with a unicast, multicast or broadcast destination address and a unicast or multicast (including broadcast) source address. Multicast source addresses have the group bit (bit 40) cleared before ALE processing begins, changing the multicast source address to a unicast source address. A multicast address of all ones is the broadcast address which may be added to the table. A learned unicast source address is added to the table with the following control bits:

**Table 9-8. Learned Address Control Bits**

Bit(s)	Value
Ageable	1
Touch	1
BLOCK	0
SECURE	0

If a received packet has a source address that is equal to the destination address then the following occurs:

- The address is learned if the address is not found in the table.
- The address is updated if the address is found.
- The packet is dropped.

### Table Entry Type

00 - Free Entry

01 - Address Entry : unicast or multicast determined by destination **address bit 40**. (1:Multicast, 0:Unicast)

10 - VLAN entry



11 - VLAN Address Entry : unicast or multicast determined by **address bit 40**.(1:Multicast, 0:Unicast)

#### 9.4.7.1.6.1 Free Table Entry

**Table 9-9. Free (Unused) Address Table Entry Bit Values**

70:62	61:60	59:0
Reserved	ENTRY_TYPE (00)	Reserved

#### 9.4.7.1.6.2 Multicast Address Table Entry (Bit 40==1)

**Table 9-10. Multicast Address Table Entry Bit Values**

70:68	67:66	65	64	63:62	61:60	59:48	47:0
Reserved	PORT_MASK	SUPER	IgnoreMbits	MCAST_FWD_STATE	ENTRY_TYPE (01)	Reserved	MULTICAST_ADDRESS

#### IgnoreMbits

Ignore Multicast Bits - Indication that the Multicast Address has ignored bits.

#### Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

#### Port Mask(1:0) (PORT\_MASK)

This 2-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

#### Multicast Forward State (MCAST\_FWD\_STATE)

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT\_MASK has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

#### Table Entry Type (ENTRY\_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

#### Packet Address (MULTICAST\_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

#### 9.4.7.1.6.3 VLAN/Multicast Address Table Entry (Bit 40==1)

**Table 9-11. VLAN/Multicast Address Table Entry Bit Values**

70:68	67:66	65	64	63:62	61:60	59:48	47:0
Reserved	PORT_MASK	SUPER	Reserved	MCAST_FWD_STATE	ENTRY_TYPE (11)	VLAN_ID	MULTICAST_ADDRESS

#### Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

#### Port Mask(1:0) (PORT\_MASK)

This 2-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

#### Multicast Forward State (MCAST\_FWD\_STATE)

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT\_MASK has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

#### Table Entry Type (ENTRY\_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

#### VLAN ID (VLAN\_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

#### Packet Address (MULTICAST\_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

#### 9.4.7.1.6.4 Unicast Address Table Entry (Bit 40==0)

**Table 9-12. Unicast Address Table Entry Bit Values**

70:68	67	66	65	64	63:62	61:60	59:48	47:0
Reserved	TRUNK	PORT_NUMBER	BLOCK	SECURE	UNICAST_TYPE (00) or (X1)	ENTRY_TYPE (01)	Reserved	UNICAST_ADDRESS

#### Trunk

Trunk Indicator -

0 - the port bits in the entry are the port number

1 - the port bits in the entry are the trunk number

### Port Number (PORT\_NUMBER)

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).

### Block (BLOCK)

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

### Secure (SECURE)

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port\_number.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source address and do not update the address (block must be zero)

### Unicast Type (UNICAST\_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

### Table Entry Type (ENTRY\_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

### Packet Address (UNICAST\_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

#### 9.4.7.1.6.5 OUI Unicast Address Table Entry

**Table 9-13. OUI Unicast Address Table Entry Bit Values**

70:64	63:62	61:60	59:48	47:24	23:0
Reserved	UNICAST_TYPE (10)	ENTRY_TYPE (01)	Reserved	UNICAST_OUI	Reserved

### Unicast Type (UNICAST\_TYPE)

This field indicates the type of unicast address the table entry contains.

- 00 - Unicast address that is not ageable.
- 01 - Ageable unicast address that has not been touched.
- 10 - OUI address - lower 24-bits are don't cares (not ageable).
- 11 - Ageable unicast address that has been touched.

#### Table Entry Type (ENTRY\_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

#### Packet Address (UNICAST\_OUI)

For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup.

#### 9.4.7.1.6.6 VLAN/Unicast Address Table Entry (Bit 40==0)

**Table 9-14. Unicast Address Table Entry Bit Values**

70:68	67	66	65	64	63:62	61:60	59:48	47:0
Reserved	TRUNK	PORT_NUMBER	BLOCK	SECURE	UNICAST_TYPE (00) or (X1)	ENTRY_TYPE (11)	VLAN_ID	UNICAST_ADDRESS

#### Trunk

Trunk Indicator -

- 0 - the port bits in the entry are the port number
- 1 - the port bits in the entry are the trunk number

#### Port Number (PORT\_NUMBER)

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).]

#### Block (BLOCK)

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

- 0 - Address is not blocked.
- 1 - Drop a packet with a matching source or destination address (secure must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

#### Secure (SECURE)

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry PORT\_NUMBER.

- 0 - Received port number is a don't care.
- 1 - Drop the packet if the received port is not the secure port for the source address and do not update the address (block must be zero)

#### Unicast Type (UNICAST\_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

#### Table Entry Type (ENTRY\_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

#### VLAN ID (VLAN\_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

#### Packet Address (UNICAST\_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

#### 9.4.7.1.6.7 VLAN Table Entry

**Table 9-15. Inner VLAN Table Entry**

70:68	67:66	65	64:62	61:60	59:48	47	46:44	43	42:26	25:24	23	22:20	19	18:2	1:0
Resvd	NoLrnM sk	Ingress Chk	000	10	iVLANI D	Resvd	Regldx	NoFrag	Reserve d	fwdutag	Reserved	Uregldx	LmtNxt Hdr	Reserve d	Member

**Table 9-16. Outer VLAN Table Entry**

70:68	67:66	65	64:62	61:60	59:48	47	46:44	43	42:26	25:24	23	22:20	19	18:2	1:0
Resvd	NoLrn Msk	Ingress Chk	010	10	oVLAN ID	Resvd	Regldx	NoFra g	Reserv ed	fwdata g	Reserve d	Uregld x	LmtNxt Hdr	Reserv ed	Member

#### Member

VLAN Member list - This field indicates which port(s) are a member of the associated VLAN.

#### LmtNxtHdr

VLAN Limit Next Header Control - Causes frames to be dropped if the Protocol/Nxt Header does not match the ALE\_NXT\_HDR register values

#### Uregldx

VLAN Unregister Multicast Index - Index into VLAN\_Mask\_Mux register array that is used to create the unregistered multicast flood mask.

#### fwdutag

VLAN Forward Untagged Egress - Causes the packet VLAN tag to be removed on egress for the specified port(s).

#### NoFrag

LAN No IPv4 Fragmented frames Control - Causes IPv4 fragmented IP frames to be dropped.

#### Regldx

VLAN Registered Multicast Index - Index into VLAN\_Mask\_Mux register array that is used to create the registered multicast flood mask.

#### iVLANID

Inner VLAN ID - This is the 12-bit Inner VLAN ID, normally referred to as C-VLANID

## oVLANID

Outer VLAN ID - This is the 12-bit Outer VLAN ID, normally referred to as S-VLANID

## IngressChk

VLAN Ingress Check - When set, if the receive port is not a member of this VLAN then the packet is dropped. This is similar to the vid\_ingress\_check bit in the ale\_port\_control register except this check is for this VLAN only (not all VLANs).

## NoLrnMsk

VLAN No Learn Mask - When a bit is set in this mask, a packet with an unknown source address received on the associated port will not be learned (i.e. When a VLAN packet is received and the source address is not in the table, the source address will not be added to the table).

**Table 9-17. IPv4 Table Entry**

70	69:65	64:62	61:60	59:32	31:0
Resvd	IgnBits	110	10	Reserved	IPv4Adr

**Table 9-18. IPv6 Table Entry High**

70:64	63	62	61:60	59:0
IgnBits	Resvd	1	10	IPv6Adr[127:68]

**Table 9-19. IPv6 Table Entry Low**

70:63	62	61:60	59:0
IPv6Adr[67:60]	1	10	IPv6Adr[59:0]

## IgnBits

Inore Bits - Indicates the number of lower address bits (IPv4 or IPv6) to be ignored starting at bit zero. Ignored bits must be zero value in the table entry

### 9.4.7.1.7 ALE Policing and Classification

The ALE has a number of configurable policer engines. Each policer engine can be used for classification and or policing. Policing is an extension of the classifier function that allows for color marking and rate limiting due to classifier bandwidth measurement thresholds. Any policer engine can be used for classification alone by not using the bandwidth limiting thresholds. The policers\_div\_8 field in the ALE\_Status register indicates the number of policers available to be used for classification or policing.

Each policer can be enabled to match on one or more of any of the below packet fields for classification. All but Port and Priority are index references to the ALE table entries.

## ALE Policing

The policing function on each policer engine is implemented as dual-counter three-color marking engine as described in the IETF RFC2698. The first counter is the committed information rate (CIR) counter and the second counter is the peak information rate (PIR) counter. The policing function can use either or both counters. Based on the counter values the packet color is determined. The color is used to determine whether the packet is dropped or forwarded. The ALE has a local feature that can drop packets regardless of queue state.

The policing rates are determined by the below equations:

$$\text{CIR policing rate in Mbit/s} = ((\text{ALE frequency in Mhz}) * \text{cir\_idle\_inc\_val}) / 32768$$

$$\text{PIR policing rate in Mbit/s} = ((\text{ALE frequency in Mhz}) * \text{pir\_idle\_inc\_val}) / 32768$$

Each policer has 10 different match operations (see below Classification). Since multiple policing entries can be hit on a single packet this provides the ability to create precise traffic stream control.

#### 9.4.7.1.7.1 ALE Classification

Each policer can be enabled to classify on one or more of any of the below packet fields for classification or policing. All but Port and Priority are index references to the ALE table entries.

- Port or Trunk Group Number
- Priority extracted from VLAN, mapped from DSCP if enabled or Default Port Priority
- Organization Network Unique identifier - ONU
- Destination Address - DA
- Source Address - SA
- Outer VLANID - S-VLANID
- Inner VLANID - C-VLANID
- Ether Type
- IP Source Address - IPSA with full CIDR masking for IPv4 and IPv6
- IP Destination Address - IPSA with full CIDR masking for IPv4 and IPv6

Multiple classifiers can match on a single packet. For example a classifier can be enabled to match on priority while another classifier could match on IP address. The ALE will return to the switch the highest classifier entries thread ID that matched with an enabled thread ID number. This could be used to further host routing of the packet.

##### 9.4.7.1.7.1.1 Classifier to CPPI Transmit Flow ID Mapping

The ALE can generate a 6-bit transmit CPPI Flow ID based on classifier matches that can be used instead of the switch default transmit Flow ID mapping. The switch default flow ID is the remapped received packet priority (0 to 7). Thread and flow ID are used interchangeably for this since there is a single hardware thread (TXST\_THREAD\_MREADY) but there are 6-bits of FLOW\_ID in the transmit CPPI INFO word 0. When enabled, the highest classifier match can map to a particular 6-bit flow ID value that is associated with the classifier. The ALE also supports an optional ALE default thread/flow ID value in the event that no classifiers match. Each thread/flow ID, including the ALE default thread/flow ID, has an enable such that the ALE default thread/Flow ID is used if enabled and if no matches occur (instead of the remapped received packet priority). If the ALE default is not enabled and no matches occur then the switch default value will be used. If multiple classifier matches occur, the highest match with a thread enable bit set will be used. The resultant flow ID has the CPSW\_P0\_FLOW\_ID\_OFFSET\_REG register value added to it to determine the actual value in the INFO 0 Flow ID field.

Three registers are used for ALE classification thread/flow ID mapping configuration (CPSW\_ALE\_THREAD\_DEF\_REG, CPSW\_ALE\_THREAD\_CTL\_REG and CPSW\_ALE\_THREAD\_VAL\_REG). The three thread mapping registers are used independently and are separate from the other ALE policing registers. The CPSW\_ALE\_THREAD\_CTL\_REG register allows the CPSW\_ALE\_THREAD\_VAL\_REG register contents to be written to the selected classifier. There is a single CPSW\_ALE\_THREAD\_DEF\_REG that is used for all classifiers. The thread mapping registers can be written or changed at any time but any packets that are already processed will not have their thread altered.

#### 9.4.7.1.8 Mirroring

The ALE supports three mirroring modes: destination port, source port and or table entry.

**Destination port mirroring** allows packets from any ingress port or trunk which ends up switching to a particular egress destination port or trunk to be mirrored to yet another egress destination port or trunk. For example any traffic from any port that is switched to port 'A' can be also mirrored to port 'B'. (MIRROR\_DP=A, MIRROR\_DEN=1h, MIRROR\_TOP=B in the ALE\_CONTROL register).

**Source port mirroring** allows packets received on any enabled ingress source port or trunk to be switched to the mirror egress port as well as the actual egress destination ports. For example traffic received on ingress port 'A' can be switched to egress port 'B' as well as the intended egress destination port.(MIRROR\_SP=1h in the ALE\_PORTn\_CONTROL register, MIRROR\_SEN=1h, MIRROR\_TOP=B in the ALE\_CONTROL register).

**Table entry mirroring** allows for any MAC Address, MAC Address with VLAN, ONU Address or VLAN entry that matches on ingress to be switched to the egress destination as well as the actual egress destination.



For example all traffic for VLAN ID of 35 can be mirrored to port 'B'. That is any traffic switched on VLAN ID of 35 will be mirrored. ({VLAN ID of 35 in ALE Table entry index=C}, MIRROR\_MIDX=C, MIRROR\_MEN=1h, MIRROR\_TOP=B)

In the event that mirrored packets are mirrored to or from a port that is also the mirror port the packet will not be duplicated or marked as a mirror packet since the packet has already been on the port as ingress or egress. The packet sent to the mirror port may have modified VLAN info based on the port and VLAN lookup table entries. The mirror port need not be a member of the VLAN ID it is mirroring, the ALE will forward traffic to the mirror port after ingress and egress filters are applied.

The switch may decide to drop any mirror traffic based on switch buffer thresholds as to prevent required traffic from becoming congested.

Port mirroring is controlled by register fields in ALE\_CONTROL, ALE\_CONTROL2 and the port control registers.

- MIRROR\_DP - The destination port that will have its traffic mirrored (ALE\_CONTROL\_REG register).
- MIRROR\_TOP - The port to which mirrored traffic is sent (ALE\_CONTROL\_REG register).
- MIRROR\_MEN - The enable for mirroring traffic that matches a supported lookup table entry (ALE\_CONTROL\_REG register).
- MIRROR\_DEN - The Enable for destination port mirroring (ALE\_CONTROL\_REG register).
- MIRROR\_SEN - The Enable for source port mirroring (ALE\_CONTROL\_REG register).
- MIRROR\_MIDX - The index of a lookup table entry that will be mirrored ALE\_CONTROL2\_REG register).
- Px\_MIRROR\_SP - The enable for the Source port to be mirrored. Although multiple source ports can be mirrored concurrently, a mirror traffic bandwidth issue may occur on the mirror egress port (ALE\_PORTn\_CONTROL register).

#### 9.4.7.1.9 Trunking

The ALE supports port trunking of any port in any of four trunk groups. That is, four trunk groups can be supported with up to eight ports in each trunk group. There are no port adjacency rules for trunk groups. When ports are a member of a trunk group, addresses added and used in the lookup table will refer to the trunk group rather than port as indicated in the lookup table entries. If ports are removed from a trunk group, the ALE will redistribute the traffic based on the crc polynomial of enabled fields and the remaining ports within the trunk group. A trunk group may contain only one port. Packet priority, DA, SA, C-VLAN ID, IPv4SA, IPv4DA, IPv6SA, and/or IPv6DA can be used in the hash to generate destination port within the trunk group. If all hash enables are disabled, the packet can be directed to a particular port within the trunk group which allows for testing paths etc. A host directed frame is directed to the directed port regardless of trunk group settings.

Trunking is controlled through fields in the ALE\_CONTROL2\_REG register and in each ALE\_ALE\_PORTn\_CONTROL\_REG register:

- TRK\_EN\_DST - Enable destination address hashing for trunk port calculation.
- TRK\_EN\_SRC - Enable source address hashing for trunk port calculation.
- TRK\_EN\_PRI - Enable priority hashing for trunk port calculation.
- TRK\_EN\_VLAN - Enable inner C-VLAN ID hashing for trunk port calculation.
- TRK\_EN\_SIP - Enable source IP address hashing for trunk port calculation.
- TRK\_EN\_DIP - Enable destination IP address hashing for trunk port calculation.
- TRK\_BASE - Hashing formula starting value and test port offset.
- TRUNK\_EN - Enable this port as a trunk group
- TRUNK\_NUMBER - Trunk group number defines this port as a member of a particular trunk group.

#### 9.4.7.1.10 DSCP

The ALE can map DSCP field to priority prior to classification matching. When enabled the DSCP is mapped via 64 priority entries such that any DSCP value can be mapped to any of the eight priorities. When a packet is received without a VLAN priority this remapped priority can be used instead of the default Port VLAN priority field. See P0\_RX\_DSCP\_MAP\_REG and PN\_RX\_DSCP\_MAP\_REG registers in the Register Manual section for DSCP mapping.



#### 9.4.7.1.11 Packet Forwarding Processes

There are four processes that an incoming received packet may go through to determine packet forwarding. The processes are *Ingress Filtering*, *VLAN\_Aware Lookup*, *VLAN\_Unaware Lookup*, and *Egress*.

Packet processing begins in the Ingress Filtering process. Each port has an associated packet forwarding state that can be one of four values (Disabled, Blocked, Learning, or Forwarding). The default state for all ports is Disabled. The host sets the packet forwarding state for each port.

In the packet ingress process (receive packet process), there is a forward state test for unicast destination addresses and a forward state test for multicast addresses. The multicast forward state test indicates the port states required for the receiving port in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state for the packet to be forwarded for transmission. The MCAST\_FWD\_STATE indicates the required port state for the receiving port as indicated in the preceding table. The unicast forward state test indicates the port state required for the receiving port in order to forward the unicast packet. The transmit port must be in the Forwarding state in order to forward the packet. The BLOCK and SECURE bits determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state. The transmit port must be in the Forwarding state regardless. The forward state test used in the ingress process is determined by the destination address packet type (multicast/unicast).

In general, packets received with errors are dropped by the address lookup engine without learning, updating, or touching the address. The error condition and the abort are indicated by the Ethernet port to the ALE. Packets with errors may be passed to the host (not aborted) by a Ethernet port, if the port has the RX\_CMF\_EN, RX\_CEF\_EN, or RX\_CSF\_EN bit(s) set in the CPSW\_PN\_MAC\_CONTROL\_REG register. Error packets that are passed to the host by the Ethernet port are considered to be bypass packets by the ALE and are sent only to the host. Error packets do not learn, update, or touch addresses regardless of whether they are aborted or sent to the host. Packets with long or short errors received by the host are dropped. Packets with errors received by the host are forwarded as normal.

The following control bits are in the CPSW\_PN\_MAC\_CONTROL\_REG register:

- [22] RX\_CEF\_EN - enables frames that are fragments, long, jabber, CRC, code, and alignment errors to be forwarded
- [23] RX\_CSF\_EN - enables short frames to be forwarded
- [24] RX\_CMF\_EN - enables MAC control frames to be forwarded.

##### 9.4.7.1.11.1 Ingress Filtering Process

Condition and action
If ((ALE BYPASS) and (host port is not the receive port)) then use host portmask and go to Egress process
if (directed packet) then use directed port number and go to Egress process
If (Rx PORT_STATE is Disabled) then discard the packet
if ((ALE BYPASS or error packet) and (host port is not the receive port)) then use host portmask and go to Egress process
if (((BLOCK) and (unicast source address found)) or ((BLOCK) and (unicast destination address found))) then discard the packet
if ((ENABLE_RATE_LIMIT) and (rate limit exceeded) and (not RATE_LIMIT_TX)) then if (((Multicast/Broadcast destination address found) and (not SUPER)) or (Multicast/Broadcast destination address not found)) then discard the packet

if ((not forward state test valid) and (destination address found)) then discard the packet to any port not meeting the requirements <ul style="list-style-type: none"> <li>Unicast destination addresses use the unicast forward state test and multicast destination addresses use the multicast forward state test.</li> </ul>
if ((destination address not found) and ((not transmit port forwarding) or (not receive port forwarding))) then discard the packet to any ports not meeting the above requirements
if (source address found) and (secure) and (not block) and (receive port number != port_number)) then discard the packet
if ((not super) and (drop_untagged) and ((non-tagged packet) or ((priority tagged) and not(en_vid0_mode))) then discard the packet
If (VLAN_Unaware) CPSW_ALE_FORCE_UNTAGGED_EGRESS_REG = "000000" CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG = "111111" CPSW_ALE_UNKNOWN_REG_MCAST_FLOOD_REG = "111111" VLAN_MEMBER_LIST = "111111" else if (VLAN not found) CPSW_ALE_FORCE_UNTAGGED_EGRESS_REG = CPSW_ALE_FORCE_UNTAGGED_EGRESS_REG CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG = CPSW_ALE_UNKNOWN_REG_MCAST_FLOOD_REG CPSW_ALE_UNKNOWN_REG_MCAST_FLOOD_REG = CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG VLAN_MEMBER_LIST = CPSW_ALE_UNKNOWN_VLAN_REG else CPSW_ALE_FORCE_UNTAGGED_EGRESS_REG = found CPSW_ALE_FORCE_UNTAGGED_EGRESS_REG CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG = found CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG CPSW_ALE_UNKNOWN_REG_MCAST_FLOOD_REG = found CPSW_ALE_UNKNOWN_REG_MCAST_FLOOD_REG VLAN_MEMBER_LIST = found VLAN_MEMBER_LIST
if ((not SUPER) and (VID_INGRESS_CHECK) and (Rx port is not VLAN member)) then discard the packet
if ((ENABLE_AUTH_MODE) and (source address not found) and not(destination address found and (SUPER))) then discard the packet
if (destination address equals source address) then discard the packet
if (VLAN_AWARE) goto VLAN_Aware_Lookup process else goto VLAN_Unaware_Lookup process

#### 9.4.7.1.11.2 VLAN\_Aware Lookup Process

Condition and action
if ((unicast packet) and (destination address found with or without VLAN) and (not SUPER)) then portmask is the logical "AND" of the PORT_NUMBER and VLAN_MEMBER_LIST less the host port and goto Egress process
if ((unicast packet) and (destination address found with or without VLAN) and (not SUPER)) then portmask is the logical "AND" of the PORT_NUMBER and the VLAN_MEMBER_LIST and goto Egress process

if ((unicast packet) and (destination address found with or without VLAN) and (SUPER)) then portmask is the PORT_NUMBER and goto Egress process
if (Unicast packet) # destination address not found then portmask is VLAN member LIST less host port and goto Egress process
if ((Multicast packet) and (destination address found with or without VLAN) and (not SUPER)) then portmask is the logical "AND" of CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG and found destination address/VLAN portmask (PORT_MASK) and VLAN_MEMBER_LIST and goto Egress process
if ((Multicast packet) and (destination address found with or without VLAN) and (SUPER)) then portmask is the PORT_MASK and goto Egress process
if (Multicast packet) # destination address not found then portmask is the logical "AND" of CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG and VLAN_MEMBER_LIST then goto Egress process
if (Broadcast packet) then use found VLAN_MEMBER_LIST and goto Egress process

#### 9.4.7.1.11.3 VLAN\_Unaware Lookup Process

Condition and action
if ((unicast packet) and (destination address found with or without VLAN) and (not SUPER)) then portmask is the VLAN_MEMBER_LIST less the host port and goto Egress process
if ((unicast packet) and (destination address found with or without VLAN) and (not SUPER)) then portmask is the logical "AND" of the PORT_NUMBER and the VLAN_MEMBER_LIST and goto Egress process
if ((unicast packet) and (destination address found with or without VLAN) and (SUPER)) then portmask is the PORT_NUMBER and goto Egress process
if (Unicast packet) # destination address not found then portmask is VLAN_MEMBER_LIST less host port and goto Egress process
if ((Multicast packet) and (destination address found with or without VLAN) and (not SUPER)) then portmask is the logical "AND" of CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG and found destination address/VLAN portmask (PORT_MASK) and VLAN_MEMBER_LIST and goto Egress process
if ((Multicast packet) and (destination address found with or without VLAN) and (SUPER)) then portmask is the PORT_MASK and goto Egress process
if (Multicast packet) # destination address not found then portmask is the logical "AND" of CPSW_ALE_UNKNOWN_MCAST_FLOOD_REG and VLAN_MEMBER_LIST then goto Egress process
if (Broadcast packet) then use found VLAN_MEMBER_LIST and goto Egress process

#### 9.4.7.1.11.4 Egress Process

Condition and action
Clear Rx port from portmask (don't send packet to Rx port).
Clear disabled ports from portmask.
if ((ENABLE_OUI_DENY) and (OUI source address not found) and (not ALE BYPASS) and (not error packet) and not ((mcast destination address) and (SUPER))) then Clear host port from portmask
if ((not ENABLE_OUI_DENY) and (OUI source address found) and (not ALE BYPASS) and (not error packet) and not ((mcast destination address) and (SUPER))) then Clear host port from portmask
if ((ENABLE_RATE_LIMIT) and (RATE_LIMIT_TX)) then if (not SUPER) and (rate limit exceeded on any tx port) then clear rate limited tx port from portmask If address not found then SUPER cannot be set.
If portmask is zero then discard packet
Send packet to portmask ports.

#### 9.4.7.1.11.5 Learning/Updating/Touching Processes

The learning, updating, and touching processes are applied to each receive packet that is not aborted. The processes are concurrent with the packet forwarding process. In addition to the following, a packet must be received without error in order to learn/update/touch an address.

##### 9.4.7.1.11.5.1 Learning Process

The learning process is applied to each receive packet that is not aborted. The learning process is a concurrent process with the packet forwarding process.

Condition and action
If (directed) then do not learn, update, or set touched else continue
If (not (Learning or Forwarding) or (ENABLE_AUTH_MODE) or (packet error) or (NO_LEARN)) then do not learn address
if ((Non-tagged packet) and (DROP_UNTAGGED)) then do not learn address
if ((VLAN_AWARE) and (VLAN not found) and (unknown VLAN_MEMBER_LIST = "000")) then do not learn address
if ((VID_INGRESS_CHECK) and (Rx port is not VLAN member) and (VLAN found)) then do not learn address
if ((source address found) and (receive port_number != PORT_NUMBER) and (SECURE or BLOCK)) then do not update address else continue
if ((source address found) and (receive port number != PORT_NUMBER)) then update address else continue
if ((source address not found) and (VLAN_AWARE) and not (LEARN_NO_VID)) then learn address with VLAN
if ((source address not found) and ((not VLAN_AWARE) or (VLAN_AWARE and LEARN_NO_VID))) then learn address without VLAN

#### 9.4.7.1.11.5.2 Updating Process

Condition and action
if (dlr_unicast) then do not update address
If (not(Learning or Forwarding) or (ENABLE_AUTH_MODE) or (packet error) or (NO_SA_UPDATE)) then do not update address
if ((Non-tagged packet) and (DROP_UNTAGGED)) then do not update address
if ((VLAN_AWARE) and (VLAN not found) and (unknown VLAN_MEMBER_LIST = "000")) then do not update address
if ((VID_INGRESS_CHECK) and (Rx port is not VLAN member) and (VLAN found)) then do not update address
if ((source address found) and (receive port number != PORT_NUMBER) and (SECURE or BLOCK)) then do not update address
if ((source address found) and (receive port number != PORT_NUMBER)) then update address

#### 9.4.7.1.11.5.3 Touching Process

if ((source address found) and (ageable) and (not touched)) then set touched
---

### 9.4.7.2 CPPI CPDMA Host Interface

#### 9.4.7.2.1 Functional Operation

For legacy reasons this document uses FHost ("cpsw ingress **From Host**") interchangeably with host receive and THost ("cpsw egress **To Host**") interchangeably with host transmit.

Host Software sends and receives network frames via the CPDMA CPPI 3.0 compliant host interface. The host interface includes module registers and host memory data structures. The host memory data structures are buffer descriptors and data buffers. Buffer descriptors are data structures that contain information about a single data buffer. Buffer descriptors may be linked together to describe frames or queues of frames for transmission of data from the host to Ethernet and free buffer queues available for packet data from Ethernet to the host.

After reset, initialization, and configuration the host may initiate CPDMA host interface operations. FHost DMA operations (from host to Ethernet) are initiated by host writes to the appropriate FHost channel head descriptor pointer. The FHost DMA controller then fetches the first packet in the packet chain from memory in accordance with CPPI 3.0 protocol and proceeds with packet operations. The DMA controller fetches the packet data in 64-byte (maximum) bursts.

Host CPDMA THost operations are initiated by host writes to the appropriate THost channel head descriptor pointer after host initialization and configuration. The THost DMA controller writes Ethernet received packet data to external host memory in accordance with CPPI 3.0 protocol.

#### 9.4.7.2.2 THost CPDMA Interface

The THost DMA (Ethernet to host) is an eight channel CPPI 3.0 compliant interface. Each priority/channel (priority and channel are used interchangeably) has a single queue for frame reception.

##### 9.4.7.2.2.1 THost CPDMA Host Configuration

To configure the CPDMA for THost operations the host must perform the following:

- Initialize the CPDMA\_TH(0..7)\_HDP Registers to zero.
- Enable the desired THost interrupts in the CPDMA\_TH\_INTSTAT\_SET register.

- Write the CPDMA\_TH\_BUFFER\_OFFSET register value.
- Setup the channel(s) buffer descriptors in host memory as required by CPPI.
- Enable the CPDMA controller by setting the TH\_EN bit in the CPDMA\_TH\_CONTROL register.

#### 9.4.7.2.2.2 THost CPDMA Buffer Descriptors

A THost buffer descriptor is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

**Table 9-20. THost Buffer Descriptor format**

Word Offset	Bit fields																																	
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
	next descriptor pointer																																	
	buffer pointer																																	
	reserved				buffer offset												reserved				buffer length													
	sop	pop	own	eq	et	pass	long	short	mask	overflow	pk	et	rl	om	nc	rt	ts	em	cc	rh	es	cr	ve	de	com	plete	packet length							

Field	Description
next_descriptor_pointer	Next Descriptor Pointer - The 32-bit word aligned memory address of the next buffer descriptor in the RX queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the <b>next_descriptor_pointer</b> .
buffer_pointer	Buffer Pointer - The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the <b>buffer_pointer</b> .
buffer_offset	Buffer Offset – Indicates how many unused bytes are at the start of the buffer. The buffer offset is reduced to 12-bits. A value of 0x0000 indicates that there are no unused bytes at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The port writes the <b>buffer_offset</b> with the value from the <b>CPDMA_TH_BUFFER_OFFSET</b> register value. The host initializes the <b>buffer_offset</b> to zero for free buffers. The <b>buffer_length</b> must be greater than the <b>CPDMA_TH_BUFFER_OFFSET</b> register value. The buffer offset is valid only on <b>sop</b> .
buffer_length	Buffer Length – Indicates how many valid data bytes are in the buffer. The buffer length is reduced to 12-bits. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host initializes the <b>buffer_length</b> , but the port may overwrite the host initiated value with the actual buffer length value on SOP and/or EOP buffer descriptors. SOP buffer length values will be overwritten if the packet size is less than the size of the buffer or if the offset is nonzero. EOP buffer length values will be overwritten if the entire buffer is not filled up with data. The <b>buffer_length</b> must be greater than zero.

Field	Description
sop	Start of Packet - Indicates that the descriptor buffer is the first buffer in the packet. The port sets the <b>sop</b> bit. 0 - Not start of packet buffer 1 - Start of packet buffer
eop	End of Packet - Indicates that the descriptor buffer is the last buffer in the packet. The port sets the <b>eop</b> bit. 0 - Not end of packet buffer. 1 - End of packet buffer.
ownership	Ownership - Indicates ownership of the packet and is valid only on <b>sop</b> . This bit is set by the host and cleared by the port when the packet has been transferred. The host uses this bit to reclaim buffers. 0 - The packet is owned by the host 1 - The packet is owned by the port
eoq	End Of Queue - Set by the port to indicate that the RX queue empty condition exists. This bit is valid only on <b>eop</b> . The port determines the end of queue condition by a zero <b>next_descriptor_pointer</b> . 0 - The RX queue has more buffers available for reception. 1 - The Descriptor buffer is the last buffer in the last packet in the queue.
teardown_complete	Teardown Complete - Set by the port to indicate that the host commanded teardown process is complete, and the channel buffers may be reclaimed by the host. This bit is valid only on <b>sop</b> . 0 - The port has not completed the teardown process. 1 - The port has completed the commanded teardown process.
passed_crc	Passed CRC - Set by the port to indicate that the CRC was passed with the data. The <b>Packet_Length</b> includes the CRC bytes. The <b>passed_crc</b> bit is valid only on SOP. The <b>p0_tx_crc_remove</b> bit in the <b>CPDMA_Control</b> register determines if CPPI THost packets have a CRC included or not.
long	Jabber Frame - Indicates that the frame is a jabber frame and was not discarded because <b>rx_cef_en</b> was set in the ingress port <b>Pn_MAC_Control</b> register. Valid only on SOP.
short	Fragment Frame - Indicates that the frame is a fragment and was not discarded because <b>rx_cef_en</b> was set in the ingress port <b>Pn_MAC_Control</b> register. Valid only on SOP.
mac_ctl	Control Frame - Indicates that the frame is a MAC control frame and was not discarded because the <b>rx_cmf_en</b> bit was set in the ingress port <b>Pn_Mac_Control</b> register. Valid only on SOP.
overrun	Overrun - Set by the port to indicate that the frame reception was aborted due to THost buffer overrun. This bit is valid only on SOP. 0 - no overrun occurred on the packet 1 - The packet was aborted due to overrun
pkt_error	Packet Contained Error on Ethernet Ingress. This field is valid on SOP. - 00 - no error 01 - CRC error on ingress 10 - Code error on ingress 11 - align error on ingress
vlan_encap	VLAN Encapsulated Packet - Indicates when set that the packet data contains a 32-bit VLAN header word that is included in the packet byte count. This field is set by the port to be the value of the <b>CPDMA_Control</b> register <b>th_vlan_encap</b> bit. If both <b>th_vlan_encap</b> and <b>th_ts_encap</b> are set then the VLAN is first. This encapsulated word also contains the ALE classification FLOW (threadval). This bit is valid on SOP.
ts_encap	Timestamp Encapsulated Packet - Indicates when set that the packet data contains a 64-bit timestamp (two 32-bit words with the lower 32-bit word first) that is included in the packet byte count. This field is set by the port to be the value of the <b>CPDMA_Control</b> register <b>th_ts_encap</b> bit. If both <b>th_vlan_encap</b> and <b>thost_ts_encap</b> are set then the VLAN is first. This bit is valid on SOP.



Field	Description
chksum_encap	Checksum Encapsulated Packet – Indicates when set that the packet data contains 4-bytes of THost checksum information at the end of the packet (last 4 bytes). The packet length includes the checksum bytes.
memory_protect_error	Memory Protect Error – An error was detected in the packet Castagnoli protect CRC. The Packet should be dropped by the host.
from_port	From Port – Indicates the Ethernet ingress port number. This field is valid only on SOP.
packet_length	Packet Length – Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the buffer_length fields should equal the packet_length. Valid only on SOP.

#### 9.4.7.2.2.3 THost CPDMA Channel Teardown

The host commands a THost channel teardown by writing the channel number to the **CPDMA\_TH\_TEARDOWN** register. When a teardown command is issued to an enabled THost channel the following will occur:

- Any current frame in reception will complete normally.
- The teardown\_complete bit will be set in the next THost buffer descriptor in the chain if there is one.
- The channel head descriptor pointer will be cleared to zero
- A THost interrupt for the channel will be issued to the host.
- The host should acknowledge a teardown interrupt with a 0xffffffff acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a 0xffffffff acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be 0xffffffff if the interrupt was due to a teardown command.

#### 9.4.7.2.3 FHost CPDMA Interface

The FHost DMA is an eight channel CPPI 3.0 compliant interface. Priority between the eight queues may be either fixed or round robin as selected by **FH\_PTYPE** in the **CPDMA\_Control** register. If the priority type is fixed, then channel 7 has the highest priority and channel 0 has the lowest priority. Round robin priority proceeds from channel 0 to channel 7. Packet Data transfers occur on the TX\_VBUSP interface in 64-byte maximum burst transfers. Any packet can be designated by the host to generate a host timesync event on Ethernet egress by setting the **host\_event** bit in the packet buffer descriptor.

##### 9.4.7.2.3.1 FHost CPDMA Host Configuration

To configure the CPDMA for FHost operations the host must do the following:

- Initialize the CPDMA\_FH(0..7)\_HDP registers to a zero value.
- Enable the desired ingress interrupts in the CPDMA\_FH\_INTSTAT\_MASKED\_SET register.
- Setup the transmit channel(s) buffer descriptors in host memory as defined in CPPI 3.0.
- Configure and enable the ingress operation as desired in the CPDMA\_FH\_Control register.
- Write the appropriate CPDMA\_FH(0..7)\_HDP registers with the appropriate values to start packet operations.

##### 9.4.7.2.3.2 FHost CPDMA Buffer Descriptors

An FHost buffer descriptor is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

**Table 9-21. FHost Buffer Descriptor format**

W o r d O f f s e t	Bit fields																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	next descriptor pointer																															
1	buffer pointer																															
2	buffer offset																buffer length															



**Table 9-21. FHost Buffer Descriptor format (continued)**

3	s o p	e o p	o w n e r s h i p	e o q	t e a s r d o w n — c o m p l e t e	p a s s _ c r c	c r c _ t y p e	r e s e r v e d		t o _ p o r t _ e n	t o _ p o r t		h o s t _ e v e n t	c h k s _ u s e r _ e n c a p	r e s e r v e d	packet_length
---	-------------	-------------	---	-------------	--	--------------------------------------	--------------------------------------	--------------------------------------	--	--	---------------------------------	--	--	---	--------------------------------------	---------------

Name	Description
next_descriptor_pointer	Next Descriptor Pointer - The 32-bit word aligned memory address of the next buffer descriptor in the FHost queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the <b>next_descriptor_pointer</b> .
buffer_pointer	Buffer Pointer - The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the <b>buffer_pointer</b> .
buffer_offset	Buffer Offset – Indicates how many unused bytes are at the start of the buffer. A value of 0x0000 indicates that no unused bytes are at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The host sets the <b>buffer_offset</b> value (which may be zero to the buffer length minus 1). Valid only on <b>sop</b> .
buffer_length	Buffer Length – Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host sets the <b>buffer_length</b> . The <b>buffer_length</b> must be greater than zero.
sop	Start of Packet - Indicates that the descriptor buffer is the first buffer in the packet. 0 - Not start of packet buffer 1 - Start of packet buffer
eop	End of Packet - Indicates that the descriptor buffer is the last buffer in the packet. 0 - Not end of packet buffer. 1 - End of packet buffer.
ownership	Ownership - Indicates ownership of the packet and is valid only on <b>sop</b> . This bit is set by the host and cleared by the port when the packet has been transferred. The host uses this bit to reclaim buffers. 0 - The packet is owned by the host 1 - The packet is owned by the port
eqq	End Of Queue - Set by the port to indicate that all packets in the queue have been transferred and the FHost queue is empty. End of queue is determined by the port when the <b>next_descriptor_pointer</b> is zero on an <b>eop</b> buffer. This bit is valid only on <b>eop</b> . 0 - The FHost queue has more packets to transfer. 1 - The Descriptor buffer is the last buffer in the last packet in the queue.

Name	Description
teardown_complete	Teardown Complete – Set by the port to indicate that the host commanded teardown process is complete, and the channel buffers may be reclaimed by the host. This bit is valid only on <b>sop</b> . 0 - The port has not completed the teardown process. 1 - The port has completed the commanded teardown process.
crc_type	CRC Type – 0 – Ethernet CRC 1 – Castagnoli CRC (if <b>\$CPPI_Cast</b> = 1)
pass_crc	Pass CRC – Valid only on SOP 0 – A CRC is not included with the packet data. The Ethernet port(s) will generate the CRC on Ethernet egress. A CRC (or placeholder) at the end of the data is allowed, but not required, and the <b>buffer_count</b> and <b>packet_length</b> fields should not include the CRC bytes if they are present. 1 – A CRC is included with the host packet data. The <b>packet_length</b> and <b>buffer_count</b> fields should include the four CRC bytes. The host supplied CRC should be in the last four bytes of the data.
to_port	To Port – Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode. 1 – Send the packet to port 1 if <b>to_port_en</b> is asserted. 2 – Send the packet to port 2 if <b>to_port_en</b> is asserted.
to_port_en	To Port Enable – Indicates when set that the packet is a directed packet to be sent to the <b>to_port</b> field port number. This field is set by the host. The packet is sent to one port only (index not mask). This bit is valid on SOP. 0 – not a directed packet 1 – directed packet
host_event	Host Timesync Event – Generate a host timesync event on Ethernet egress. The upper 28-bits of the packet SOP buffer descriptor address are the domain[7:0], message_type[3:0], and sequence_id[15:0] in that order. 0 – The packet will not generate a host event on Ethernet egress 1 – The packet will generate a host event on Ethernet egress
chksum_encap	Checksum Encapsulated Packet – Indicates when set that the packet data contains 4-bytes of FHost checksum information at the start of the packet (first 4 bytes). The packet length includes the checksum bytes.
packet_length	Packet Length – Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the buffer_length fields should equal the packet_length. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths.

#### 9.4.7.2.3.3 FHost CPDMA Channel Teardown

The host commands a FHost channel teardown by writing the channel number to the **CPDMA\_FH\_Teardown** register. When a teardown command is issued to an enabled FHost channel the following will occur:

- Any frame currently in transmission will complete normally
- The teardown complete bit will be set in the next sop buffer descriptor (if there is one).
- The channel head descriptor pointer will be set to zero.
- An interrupt will be issued to inform the host of the channel teardown.
- The host should acknowledge a teardown interrupt with a 0xffffffff acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due

to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a 0xffffffff acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be 0xffffffff if the interrupt was due to a teardown command.

#### 9.4.7.2.4 VLAN Aware Mode

The CPSW is in VLAN aware mode when the CPSW Control register **vlan\_aware** bit is set. In VLAN aware mode port 0 THost packets may or may not be VLAN encapsulated depending on the CPDMA\_Control register **th\_vlan\_encap** bit. The header packet VLAN is generated as described in later sections of this specification. VLAN encapsulated receive packets have a 32-bit VLAN header encapsulation word added to the packet data. VLAN encapsulated packets are specified by a set **vlan\_encap** bit in the packet buffer descriptor. The VLAN encapsulation header is included in the packet length and has the below format:

32-bit VLAN Header Encapsulation Word																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
hdr				pkt				type				reserved																			

Field	Description
hdr_pkt_priority	Header Packet VLAN priority (7 is highest priority)
hdr_pkt_cfi	Header Packet VLAN CFI bit. The
hdr_pkt_vid	Header Packet VLAN ID.
flow	FLOW – A nonzero value indicates that the ALE matched a classifier with the <b>flow</b> (threadval).
pkt_type	Packet Type – Indicates whether the packet is a VLAN tagged, priority tagged or non-tagged packet. 00 – VLAN tagged packet 01 – reserved 10 – priority tagged packet 11 – non-tagged packet

#### 9.4.7.2.5 VLAN Unaware Mode

The CPSW is in VLAN unaware mode when the CPSW Control register **vlan\_aware** bit is cleared. Port 0 THost packets (egress) may or may not be VLAN encapsulated depending on the CPSW Control register **th\_vlan\_encap** bit.

#### 9.4.7.2.6 CPDMA Command IDLE

The **cmd\_idle** bit in the **CPDMA\_Control** register allows CPDMA operation to be suspended. When the idle state is commanded, the CPDMA will stop processing THost and FHost frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For FHost, any frame in process will be completed. For THost, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Commanded idle is similar in operation to emulation control and clock stop.

#### 9.4.7.2.7 CPDMA CPPI 3.0 Interface Bandwidth

The HOST CPPI 3.0 FHost and THost interfaces are capable of supporting line rate on the Ethernet ports provided that the clock frequency is sufficient, and provided that the Host master VBUSP read/write latency is low.

#### 9.4.8 CPPI Checksum Offload

The CPPI host port can be enabled to perform checksum offload on host port packet ingress and egress. UDP and TCP over IPV4 and IPV6 are supported. For the purposes of checksum description, the first packet byte (the first byte of the destination address) is byte 1 (not byte 0). That is, a 64 byte packet goes from byte 1 to byte 64. For all packet types, the **s\_cn\_switch** bit must be set for the **vlan\_ltype\_outer** to be supported. Because it's not now a dual vlan switch, A C-switch cannot have an outer VLAN. An S-switch can have an inner, an outer, or both (outer then inner).

##### 9.4.8.1 CPPI THost Checksum Offload

When **p0\_tx\_chksum\_en** is set in **P0\_Control**, IPV4 and IPV6 UDP and TCP packets received on any Ethernet port and destined for port 0 egress are checked for correct checksum as described below. The EOP THost buffer descriptor bit **chksum\_encap** indicates whether or not the THost checksum information is included with the THost egress packet or not. If the checksum information is included in the packet the **packet\_length** includes the four checksum information bytes. The byte counts below are shown for packets with no VLAN's. The byte counts vary with one or two packet VLANs. Packets received on an Ethernet port with errors are not checked for a correct checksum if they are passed to the host (no checksum information with the error packet).

##### 9.4.8.1.1 IPV4 UDP

- Byte 15 Upper Nibble = 4 for IPV4
- Byte 15 Lower Nibble = IHL - Nibble with number of 32-bit words in IPV4 header (5 to 15 supported).
- Bytes 20:21 = fragment[15:0] – Bit 13 is the MF bit and bits 12:0 are the Fragment offset. A packet is a fragment if the MF bit is set or if the fragment offset is non-zero. The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets have MF=0 and a zero offset. A count is output for packet fragments but no errors are reported. First fragments have the UDP header included in the count. Middle and last fragments have only data included in the count (there is no UDP header).
- Byte 24 = 0x11 for UDP protocol.
- Received packet UDP checksum of zero means that there is no IPV4 checksum sent with the packet so no error will be issued.
- Received packet UDP checksum of 0xffff means that the checksum was calculated to be 0xffff or 0x0000 but was sent in the transmitted packet as 0xffff by the sending originating entity.

##### 9.4.8.1.2 IPV4 TCP

- Byte 15 Upper Nibble = 4 for IPV4
- Byte 15 Lower Nibble = IHL - Nibble with number of 32-bit words in IPV4 header (5 to 15 supported).
- Bytes 20:21 = fragment[15:0] – Bit 13 is the MF bit and bits 12:0 are the Fragment offset. A packet is a fragment if the MF bit is set or if the fragment offset is non-zero. The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets have MF=0 and a zero offset. A count is output for packet fragments but no errors are reported. First fragments have the UDP header included in the count. Middle and last fragments have only data included in the count (there is no TCP header).

- Byte 24 = 0x06 for TCP protocol.

#### 9.4.8.1.3 IPV6 UDP

- Byte 15 upper nibble = 6 for IPV6.
- Byte 21 = 0x11 for UDP protocol as next header.
- Fragment extension headers are supported. First fragments have a fragment extension header (byte 21 = 0x2c) followed by a UDP header (byte 55 = 0x11). Middle and last fragments have a fragment extension header followed by data only (no UDP header). The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets do not have a fragment extension header. A count is output for packet fragments but no errors are reported.
- Received packet UDP checksum of zero means that there is no IPV6 checksum sent with the packet so no error will be issued.
- Received packet UDP checksum of 0xffff means that the checksum was calculated to be 0xffff or 0x0000 but was sent in the transmitted packet as 0xffff by the sending originating entity.

#### 9.4.8.1.4 IPV6 TCP

- Byte 15 upper nibble = 6 for IPV6.
- Byte 21 = 0x06 for TCP protocol as next header.
- Fragment extension headers are supported. First fragments have a fragment extension header (byte 21 = 0x2c) followed by a UDP header (byte 55 = 0x06). Middle and last fragments have a fragment extension header followed by data only (no TCP header). The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets do not have a fragment extension header. A count is output for packet fragments but no errors are reported.

#### 9.4.8.1.5 THost Checksum Encapsulation Word

The 4-byte checksum encapsulation word is included as the last 4-bytes of the THost packet data when EOP buffer descriptor **chksum\_encap** is set. The **packet\_length** includes the four encapsulation bytes.

THost Checksum Encapsulation Word																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
reserved											i	i	t	f	c	checksum_add															
											p	p	c	r	h																
											v	v	p	a	k																
											4	6	_	g	s																
											_	_	u	m	u																
											v	v	d	e	m																
											a	a	p	n	_																
											l	l	_	t	e																
											i	i	n		r																
											d	d			r																
															o																
															r																

Field	Name	Description
31:21	reserved	
20	ipv4_valid	IPV4 Valid – An IPV4 TCP or UDP packet was detected
19	ipv6_valid	IPV6 Valid – An IPV6 TCP or UDP Packet was detected
18	tcp_udp_n	TCP or UDP packet – Valid only when either the <b>ipv4_valid</b> or <b>ipv6_valid</b> bits are set. 0 – Indicates UDP packet was detected. 1 – Indicates TCP packet was detected.

Field	Name	Description
17	fragment	Fragment – Indicates that an IP fragment was detected. Valid only when either the <b>ipv4_valid</b> or <b>ipv6_valid</b> bits are set.
16	checksum_error	Checksum Error detected. Valid only when either the <b>ipv4_valid</b> or <b>ipv6_valid</b> bits are set.
15:0	checksum_add	Checksum Add Value – This is the value that was summed during the checksum computation. This value is 0xffff for IPV4/6 UDP/TCP packets with no checksum error.

#### 9.4.8.2 CPPI FHost Checksum Offload

Packets sent from host port 0 (switch ingress) to any Ethernet port can have a checksum calculated and inserted into the Ethernet egress packet. The **rx\_chksum\_en** bit in the **P0\_CONTROL** register must be set for receive checksum operation to be enabled. When enabled and when the **chksum\_encap** SOP FHost buffer descriptor is set, the first four packet bytes contain the checksum information which determines how the checksum is calculated. The **checksum\_result** field determines where the checksum is inserted in the egress packet. The checksum result location is adjusted by the egress port if a VLAN is to be inserted or removed on Ethernet port egress.

##### 9.4.8.2.1 FHost Checksum Encapsulation Word

The 4-byte FHost checksum encapsulation word is included as the first four bytes of the packet data when **chksum\_encap** is set in the FHost SOP buffer descriptor. The **packet\_length** includes the four checksum encapsulation bytes.

FHost Checksum Encapsulation Word																																	
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
checksum_result								checksum_start_byte								c	r	checksum_bytecount															
																h	e																
																k	s																
																s	e																
																u	r																
																m	v																
																—	e																
																i	d																
																v																	

Field	Name	Description
31:24	<b>checksum_result</b>	Checksum Result Byte Location – This is the packet byte number where the checksum result will be placed in the egress packet. The first packet byte which is the first byte of the destination address is byte 1 (not byte zero).
23:16	<b>checksum_start_byte</b>	Checksum Start Byte – This is the packet byte number to start the checksum calculation on. The first packet byte is byte 1.
15	<b>chksum_inv</b>	Checksum Invert Zero – When set, a zero checksum value will be inverted and sent as 0xffff.
14	reserved	
13:0	checksum_bytecount	Checksum Byte Count – This is the number of bytes to calculate the checksum on. The outgoing Ethernet packet will have a checksum inserted when this value is non-zero.

#### 9.4.9 Jumbo packets

Jumbo packets are supported on the device. The size of the jumbo packet is 3024 bytes. No special configuration is required apart from appropriate length in the length fields of descriptors.

#### 9.4.10 Zero touch of Tx (From Host) buffer descriptors

The configured fields of the Tx (From Host) buffer descriptors shall not be touched by CPSW HW to enable reuse of a pre-programmed list of buffer descriptors.

This feature can be enabled by setting the 'fhost\_ownership' bit in the CPDMA\_CONTROL register.

When this is enabled the ownership bit in the Descriptor is not be touched by the HW. This enables the descriptors to be reused without much CPU intervention.

#### CPDMA\_Control - Control

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
	reserved																							f	t	t	t	t	c	t	t	f
																								h	h	h	h	h	p	h	h	h
																								o	o	o	o	o	d	o	o	o
																								s	s	s	s	s	m	s	s	s
																								t	t	t	t	t	a	t	t	t
																								—	—	—	—	—	—	—	—	—
																								o	c	t	v	c	c	o	o	p
																								w	h	s	l	e	m	f	w	t
																								n	—	—	a	f	d	f	n	y
																								e	o	e	n		—	l	e	p
																								r	v	n	—		l	e	r	e
																								s	e	c	e		d	n	s	
																								h	r	a	n		l	—	h	
																								i	r	p			e	b	i	
																								p	i	p						
																								d	e							
Re set Value	0																							0	0	0	0	0	0	0	0	0
Type	r																							rw	rw	rw	rw	rw	rw	rw	rw	rw

Bit	Name	Description
31:9	reserved	read as zero
8	fhost_ownership	FHost Ownership Write Bit Value. 0 – The CPDMA writes the FHost buffer descriptor ownership bit to zero at the end of packet processing as specified in CPPI 3.0. 1 – The CPDMA writes the FHost buffer descriptor ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each time the buffer descriptor is used. Software must set this bit when building the packet chain.

Bit	Name	Description
7	<b>thost_ch_override</b>	THost Channel Classification Match Override Enable 0 – The THost channel is not overridden with the ALE classification match. 1 – The THOST channel is overridden with the lower 3-bits of the ALE classification match value (if a classification match occurred).
6	<b>thost_ts_encap</b>	THost Packet Timestamp Encapsulation 0 – THost packets do not contain a 64-bit timestamp 1 – THost packets contain a 64-bit timestamp prepended to the packet data (32-bit lsword first).
5	<b>thost_vlan_encap</b>	THost Packet VLAN Encapsulation 0 – THost packets are not VLAN encapsulated 1 – THost packets are VLAN encapsulated
4	<b>thost_cef</b>	THost Copy Error Frames Enable – Enables THost DMA overrun frames to be transferred to memory (up to the point of buffer overrun). The overrun error bit will be set in the frame EOP buffer descriptor. Overrun frame data will be filtered when <b>thost_cef</b> is not set. THost frames with other error bits set are not affected by this bit. This is related only to frames that overrun on the THost DMA due to buffer limitations. 0 – Frames containing overrun errors are filtered. 1 – Frames containing overrun errors are transferred to memory.
3	<b>cpdma_cmd_idle</b>	CPDMA Command Idle 0 – Idle not commanded 1 – Idle Commanded (read <b>idle</b> in <b>CPDMA_Status</b> register)
2	<b>thost_offlen_block</b>	THost Offset/Length word write block. 0 – Do not block the DMA writes to the THost buffer descriptor offset/buffer length word. The offset/buffer length word is written as specified in CPPI 3.0. 1 – Block all CPDMA DMA controller writes to the THost buffer descriptor offset/buffer length words during CPPI packet processing. When this bit is set, the CPDMA will never write the third word to any THost buffer descriptor.



Bit	Name	Description
1	thost_ownership	THost Ownership Write Bit Value. 0 – The CPDMA writes the THost buffer descriptor ownership bit to zero at the end of packet processing as specified in CPPI 3.0. 1 – The CPDMA writes the THost buffer descriptor ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each time the buffer descriptor is used. Software must set this bit when building the packet chain.
0	fhost_ptype	FHost Queue Priority Type 0 – The queue uses a round robin scheme to select the next channel. 1 – The queue uses a fixed (channel 7 highest priority) priority scheme to select the next channel.

#### 9.4.11 Continuous transfer of packets from the queue with an end of transmission indication

After the completion of the transfer of all the queued packet an interrupt to the host is raised instead of at each packet transfer.

The details of enabling this is as described below.

#### Fhost Packet Completion Interrupts

The FHost DMA controller has eight channels with each channel having a corresponding interrupt (**FH\_PEND[7:0]**). The FHost interrupts are level interrupts that remain asserted until cleared by the host. Each of the eight channel interrupts may be individually enabled by setting to one the appropriate bit in the **CPDMA\_FH\_IntMask\_Set** register. Each of the eight channel interrupts may be individually disabled by clearing to zero the appropriate bit in the **CPDMA\_FH\_IntMask\_Clear** register. The raw and masked interrupt status may be read by reading the **CPDMA\_FH\_IntStat\_Raw** and **CPDMA\_FH\_IntStat\_Masked** registers respectively. When each packet transfer is complete, the CPDMA issues an interrupt to the host by writing the packet's last buffer descriptor address to the appropriate channel **CPDMA\_FH(0..7)\_CP** completion pointer register. The interrupt is generated by the write regardless of the value written. Upon interrupt reception, the host processes one or more packets from the queue and then acknowledges an interrupt by writing the address of the last buffer descriptor processed to the queue's associated Fhost Completion Pointer. If the host written buffer address value is different from the buffer address written by the port then the level interrupt remains asserted which means that the CPDMA has transferred more packets than the host has processed interrupts for. If the host written buffer address value is equal to the port written value then the host has processed all packets that the CPDMA has transferred and the level interrupt is deasserted. The CPDMA write to the completion pointer actually stores the value. The host written value is actually not written to the register location. The host written value is compared to the register contents which was written by the CPDMA and if the two values are equal then the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet. Refer to the referenced CPPI 3.0 specification for more details on interrupt processing.

#### Fhost Interrupt on EOQ

A channel packet completion will issue an interrupt only on EOQ when the corresponding **fhost\_eoq\_int[7:0]** bit is set in the **CPDMA\_FH\_Control2** register. This mode allows an entire chain of packets to be transferred before an interrupt is issued to the host (instead of interrupting at the completion of each individual packet).

#### 9.4.12 Controlled transmission of packets

The device supports to control the transmission of the packets based on an external trigger indication. The CPDMA shall fetch the next packet (from the SOP indication) only when an external trigger indication is also satisfied.

### FHost CPDMA Hardware Controlled Packet Transmission

When configured with hardware packet transmission the FHost interface can be enabled to transfer packets due to rising edges on a channel's corresponding **FH\_HW\_TRIG[7:0]** input. Each channel has a corresponding independent internal **send\_cnt[15:0]** counter. To enable the hardware controlled packet transmission for a channel, software sets the channel's corresponding bit in the **fh\_hw\_trig\_en[7:0]** field in the **CPDMA\_FH\_Control2** register. Hardware packet transmission then operates as described below:

- The channel **send\_cnt[15:0]** is cleared to zero when the channel HDP is zero (IDLE).
- Software writes the channel HDP to begin the packet chain operation.
- An asserted **FHOST\_HW\_TRIG[7:0]** input increments the associated channel **send\_cnt[15:0]** when the channel's HDP is non-zero.
- A single packet is transferred when **send\_cnt > 0** and then the **send\_cnt** is decremented.
- Go to a #a (IDLE) on EOQ which also zeroes the HDP, otherwise go to #d.

#### 9.4.13 Egress Packet Operations

Each CPSW egress port (Ethernet and Host) is capable of performing egress packet processing operations. **IntraVLAN** processing either adds, removes, or replaces VLAN information or does nothing. **InterVLAN** routing allows hardware routing between a limited number of VLANs - thereby allowing high-bandwidth or other routing operations to be offloaded from software to the CPSW (hardware). IntraVLAN processing and InterVLAN routing operations are mutually exclusive. In addition, OAM loopback allows the loopback packet source and destination addresses can be swapped on egress to facilitate OAM or generic testing operations.

##### 9.4.13.1 IntraVLAN Processing

All ports (Ethernet and CPPI) process packet VLAN's identically on cpsw egress. CPSW ingress packet VLAN's are not modified on ingress regardless of the port type or VLAN mode.

###### 9.4.13.1.1 VLAN Unaware Mode

An egress port is operating in the VLAN unaware mode when the **VLAN\_AWARE** bit in the **CPSW\_Control** register is cleared to zero. In VLAN unaware mode, transmit (egress) packets are not modified on egress (no IntraVLAN processing).

###### 9.4.13.1.2 VLAN Aware Mode

An egress port is operating in the VLAN aware mode when the **VLAN\_AWARE** bit in the **CPSW\_Control** register is set. In VLAN aware mode, transmitted packet data is changed depending on the packet type, the packet priority (**pkt\_pri**), and the VLAN information as shown in the below tables. The **vlan\_ltype\_sel** value is selected by the **s\_cn\_switch** bit in the **CPSW\_Control** register and is either the **vlan\_ltype\_inner** (0x8100 default) or **vlan\_ltype\_outer** (0x88a8 default) value. The **force\_untagged\_egress** bit comes from the Address Lookup Engine (ALE) lookup:

VLAN Aware Mode Non Tagged Transmit Packet Processing	
Insert VLAN Case	Non-tagged input packets have the header packet VLAN inserted when the <b>force_untagged_egress</b> bit in the transmit packet header (from the ALE) is de-asserted. The <b>vlan_ltype_sel</b> length/type is inserted after the source address followed by the two byte header packet VLAN. The header packet VLAN is composed of the <b>hdr_pkt_pri</b> , the <b>hdr_pkt_cfi</b> , and the <b>hdr_pkt_vid</b> . The packet length/type field is output four bytes later than it is input and is not removed or replaced.

VLAN Aware Mode Non Tagged Transmit Packet Processing	
No Change Case	Non tagged input packets are output unchanged when the <b>force_untagged_egress</b> transmit packet header bit is asserted.
VLAN Aware Mode Priority Tagged Transmit Packet Processing	
Replace PRI/VID Case	Priority tagged input packets have the packet VLAN ID (VID) and the packet priority replaced with the <b>hdr_pkt_vid</b> and the <b>hdr_pkt_pri</b> when the transmit packet header <b>force_untagged_egress</b> bit (from the ALE) is de-asserted.
Remove VLAN Case	Priority tagged input packets have the 4-byte packet VLAN information removed when the transmit packet header <b>force_untagged_egress</b> bit (from the ALE) is asserted. <b>The vlan_ltype_sel</b> length/type is removed as is the two byte packet VLAN. Input 64-67 byte priority tagged packets go out with the VLAN removed and padded to 64-bytes. The input CRC bytes are used as the pad data. Input 64-byte priority tagged packets use all four input CRC bytes as pad, input 65-byte priority tagged packets use three of the input CRC bytes as pad, and so on.
VLAN Aware Mode VLAN Tagged Transmit Packet Processing	
Replace PRI Case	VLAN tagged input packets are output with the packet priority replaced with the <b>hdr_pkt_pri</b> when the transmit packet header <b>force_untagged_egress</b> bit is de-asserted.
Remove VLAN Case	VLAN tagged input packets have the 4-byte packet VLAN information removed when the transmit packet header <b>force_untagged_egress</b> bit is asserted. <b>The vlan_ltype_sel</b> length/type is removed as is the two byte packet VLAN. Input 64-67 byte VLAN tagged packets go out with the VLAN removed and padded to 64-bytes. The input CRC bytes are used as the pad data. Input 64-byte VLAN tagged packets use all four input CRC bytes as pad, input 65-byte VLAN tagged packets use three of the input CRC bytes as pad, and so on. The output CRC is generated when the VLAN is removed.

VLAN tagged ingress packets of 64 to 67-bytes will be padded to 64-bytes on egress (Ethernet and CPPI port egress) if the VLAN is to be removed on egress.

#### 9.4.13.2 ALE Egress Opcode Operations

The Address Lookup Engine (ALE) can be setup to pass an 8-bit opcode to the destination port(s) egress packet processing logic. InterVLAN routing and OAM Source/Destination address swap are supported operation codes as shown in the below table.

ALE Egress OP Code	Egress Operation
0x00	No egress opcode (NOP)
0x01	InterVLAN (route) opcode 1
0x02	InterVLAN (route) opcode 2
0x03	InterVLAN (route) opcode 3
0x04	InterVLAN (route) opcode 4
0x05-0xFE	reserved
0xFF	DA/SA Swap opcode

##### 9.4.13.2.1 InterVLAN Routing

The CPSW is capable of InterVLAN routing with a limited number of routes (InterVLAN opcode 1 to 4). The Address Lookup Engine (ALE) determines an InterVLAN egress opcode for each packet to be VLAN routed via a classifier/policer configured for the route. Non VLAN routed packets do not have an InterVLAN (egress) opcode. The ALE classifier/policer can use the ingress packet destination address, source address, VLAN, IPDA, and/or IPSA to determine if a packet is to be VLAN routed or not. If a packet is to be routed, then the InterVLAN opcode is used on Ethernet packet egress with a total of 4 available VLAN routes (opcodes) per egress port. Each InterVLAN routing opcode contains the below fields in the opcode's associated **Pn\_InterVLAN\_OPX\_x** registers (where **x** = A to D):

Field	Description
da[47:0]	Destination Address
sa[47:0]	Source Address
vid[11:0]	VLAN ID
replace_da_sa	When set, the routed packet Destination Address is replaced with <b>da[47:0]</b> and the packet Source Address is replaced with <b>sa[47:0]</b> from InterVLAN opcode x (the selected egress opcode).
replace_vid	When set, Replace the packet VLAN ID with <b>vid[11:0]</b> from InterVLAN opcode x (the selected egress opcode).
dest_force_untagged_egress	<p>Remove the VLAN on the egress routed packet when set (associated with the destination VLAN).</p> <hr/> <p style="text-align: center;"><b>Note</b></p> <p>The ALE <b>force_untagged_egress</b> bit from the ALE is for the source VLAN and is ignored for an InterVLAN routed packet. On InterVLAN routed packets this bit is used because it is associated with the destination vlan. This <b>dest_force_untagged_egress</b> bit would not be set by software if the destination VLAN is different from the port VLAN since that might cause leaky VLANs.</p> <hr/>
decrement_ttl	<p>When set, the Time To Live (TTL) field in the header is decremented:</p> <ul style="list-style-type: none"> <li>IPV4 – Decrement the TTL byte and update the Header Checksum</li> <li>IPV6 – Decrement the Hop Limit.</li> </ul> <p>note: The ALE will send any IPv4/6 packet with a zero or one TTL field to the host. When this bit is cleared the TTL/Hop Limit fields are not checked or modified.</p>

#### 9.4.13.2.2 OAM Source and Destination Address Swap

To facilitate OAM or generic testing, the ALE can be configured to loopback packets from Ethernet receive to Ethernet Transmit. Packets destined for OAM loopback have only the source address and destination addresses swapped. No other egress processing is performed. The ALE does not perform lookups for a port in OAM loopback mode, but directed packets are transferred as normal.

#### 9.4.14 IEEE 1588 Clock Synchronization Support

The CPSW supports 1588 clock synchronization (annex D, annex E, and annex F). Ethernet GMII Transmit and receive time sync operation are supported. Time sync is double-step on egress but may be single-step on ingress.

##### 9.4.14.1 1588 Receive (Ingress) Packet Operation

There are two CPSW egress time sync interfaces for each Ethernet port. The first is the TS\_RX\_MII interface and the second is the TS\_RX\_DEC interface. Both interfaces are generated in the switch and are input to the CPTS module. There are register bits in the CPSW that control time sync operations in addition to the registers in the CPTS module. The TS\_RX\_MII interface issues a record signal along with a handle to the CPTS controller for each packet that is received. The record signal is a single clock pulse indicating that a receive packet has been detected at the associated port MII interface. The handle value is incremented with each packet and rolls over to zero after 15. There are 16 possible handle values so there can be a maximum of 16 packets “in flight” from the TS\_RX\_MII to the TS\_RX\_DEC block at any given time. A handle value is reused (not incremented) for any received packet that is shorter than about 31 octets (including preamble). Handle reuse on short packets prevents any possible overrun condition if multiple fragments are consecutively received. The TS\_RX\_MII logic is in the receive wireside clock domain. There is no decode logic in the TS\_RX\_MII to determine if the packet is a time sync event packet or not. Each received packet generates a record signal and new handle. The handle is sent to the CPTS controller with the record pulse and the handle is also sent to the TS\_RX\_DEC block along with the packet. The packet decode is performed in the TS\_RX\_DEC block. The decode function is separated

from the record function because in some systems the incoming packet can be encrypted. The decode function would be after packet decryption in those systems.

The TS\_RX\_DEC function decodes each received packet and determines if the packet meets the time sync event packet criteria. If the packet is determined to be a time sync event packet, then the time sync event is signaled to the CPTS controller via the TS\_RX\_DEC interface. If the packet is determined to be a time sync event packet and if the ALE did not drop the packet (the packet was to be sent to at least one port), then the ALE lookup is overridden and the packet is forced only to the host. The event signal is a single clock pulse indicating that the packet matched the time sync event packet criteria and that the associated packet handle, message type, and sequence ID are valid. No indication is given for received packets that do not meet the time sync event criteria. The 16-bit sequence ID is found in the time sync event packet at the sequence ID offset into the PTP message header (**pn\_ts\_seq\_id\_offset**). The 8-bit domain number is found in the time sync event packet at the domain offset into the PTP message header (**pn\_ts\_domain\_offset**). A packet is determined to be a receive event packet under the following conditions.

#### Note

TS\_RX\_DEC and TS\_RX\_MII are internal interfaces that have no registers. They are included here for description purposes only.

#### 9.4.14.1.1 Annex D (IPv4)

1. Receive annex D time sync is enabled (pn\_ts\_rx\_annex\_d\_en is set in the Pn\_TS\_Ctl register).
2. One of the sequences below is true.
  - The first packet LTYPE matches 0x0800
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches 0x0800
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the second packet LTYPE matches 0x0800
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the third packet LTYPE matches 0x0800
3. Byte 14 (the byte after the LTYPE) contains 0x45 (IP\_VERSION). Note that the byte numbering assumes that there are no VLANs. The byte number is intended to show the relative order of the bytes.
4. Byte 20 contains 0bXXX00000 (5 lower bits zero) and Byte 21 contains 0x00 (fragment offset zero)
5. Byte 22 contains 0x01 (HOP Limit = 1) if the ts\_ttl\_nonzero bit in the switch Pn\_TS\_CTL\_LTYPE2 register is zero, or byte 22 contains any value if ts\_ttl\_nonzero is set. Byte 22 is the TTL/HOP field.
6. Byte 23 contains 0x11 (Next Header UDP Fixed).
7. The ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is zero and Bytes 30 through 33 contain:
  - Decimal 224.0.1.129 and the ts\_129 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.1.130 and the ts\_130 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.1.131 and the ts\_131 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.1.132 and the ts\_132 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.0.107 and the ts\_107 bit in the Pn\_TS\_Ctl\_Ltype2 register is set

Or:

The ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is set and Bytes 30 through 33 contain any values.

8. Bytes 36 and 37 contain:
  - Decimal 0x01 and 0x3f respectively and the ts\_319 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 0x01 and 0x40 respectively and the ts\_320 bit in the Pn\_TS\_Ctl\_Ltype2 register is set.
9. The PTP message begins in byte 42.
10. The packet message type is enabled in the ts\_msg\_type\_en field in Pn\_TS\_Ctl.
11. The packet was received without error (not long/short/mac\_ctl/crc/code/align).

#### 9.4.14.1.2 Annex E (IPv6)

1. Receive annex E time sync is enabled (pn\_ts\_rx\_annex\_e\_en is set in the switch Pn\_TS\_Ctl register).

2. One of the sequences below is true.
  - The first packet LTYPE matches 0x86dd.
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches 0x86dd.
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the second packet LTYPE matches 0x86dd.
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the third packet LTYPE matches 0x86dd.
3. Byte 14 (the byte after the LTYPE) contains 0x6X (IP\_VERSION in most significant nibble).
4. Byte 20 contains 0x11 (UDP Fixed Next Header).
5. Byte 21 contains 0x01 (HOP Limit = 1) if the ts\_ttl\_nonzero bit in the switch Pn\_TS\_Ctl\_Ltype2 register is zero, or byte 21 contains any value if ts\_ttl\_nonzero is set. Byte 21 is the TTL/HOP field.
6. The ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is zero and Bytes 38 through 53 contain:
  - FF0M:0:0:0:0:0:0181 and the ts\_129 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0182 and the ts\_130 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0183 and the ts\_131 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0184 and the ts\_132 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:006B and the ts\_107 bit in the Pn\_TS\_Ctl\_Ltype2 register is set (all values above are 16-bit hex numbers with M is enabled in the ts\_mcast\_type\_en field in the Pn\_TS\_Ctl2 register).

Or:

The ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is set and Bytes 38 through 53 contain any value.

7. Bytes 56 and 57 contain (UDP Header in bytes 54 through 61):
  - Decimal 0x01 and 0x3f respectively and the ts\_319 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 0x01 and 0x40 respectively and the ts\_320 bit in the Pn\_TS\_Ctl\_Ltype2 register is set.
8. The PTP message begins in byte 62.
9. The packet message type is enabled in the ts\_msg\_type\_en field in Pn\_TS\_Ctl.
10. The packet was received without error (not long/short/mac\_ctl/crc/code/align).

#### 9.4.14.1.3 Annex F (IEEE 802.3)

1. Receive Annex F time sync is enabled (ts\_rx\_annex\_f\_en is set in the switch Pn\_TS\_Ctl register).
2. One of the sequences below is true.
  - The first packet LTYPE matches ts\_ltype1. LTYPE 1 should be used when only one time sync LTYPE is to be enabled.
  - The first packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches pn\_ts\_ltype1
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the second packet LTYPE matches pn\_ts\_ltype1
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the second packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the third packet LTYPE matches pn\_ts\_ltype1
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_rx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_rx\_vlan\_ltype2\_en is set and the third packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
3. The PTP message begins in the byte after the LTYPE.
4. The packet message type is enabled in the ts\_msg\_type\_en field in the Pn\_TS\_Ctl register.
5. The packet was received without error (not long/short/mac\_ctl/crc/code/align).



#### 9.4.14.2 1588 Transmit Packet Operation

There are two CPSW transmit time sync interfaces for each Ethernet port. The first is the TS\_TX\_DEC interface and the second is the TS\_TX\_MII interface. Both interfaces are internal to the cpsw and are input to the CPTS module.

The TS\_TX\_DEC function decodes each packet to be transmitted and determines if the packet meets the time sync event packet criteria. If the packet is determined to be a time sync event packet, then the time sync event is signaled to the CPTS controller via the TS\_TX\_DEC interface (**pn\_ts\_tx\_dec\_evnt**, **pn\_ts\_tx\_dec\_hndl[3:0]**, **pn\_ts\_tx\_dec\_msg\_type[3:0]**, **pn\_ts\_tx\_dec\_seq\_id[15:0]** and **pn\_ts\_rx\_dec\_domain[7:0]**). The event signal is a single clock pulse indicating that the packet matched the time sync event packet criteria and that the associated packet handle, message type, and sequence ID are valid. The 16-bit sequence ID is found in the time sync event packet at the sequence ID offset into the message header (**pn\_ts\_seq\_id\_offset**). The 8-bit domain number is found in the time sync event packet at the domain offset into the PTP message header (**pn\_ts\_domain\_offset**). No indication is given for transmit packets that do not meet the time sync event criteria. The time sync event packet handle is also passed along with the packet to the TS\_TX\_MII with an indication that the packet is a time sync event packet. Unlike receive; only transmit event packets increment the handle value. The decode function is separated from the record function because some systems may encrypt the packet. The encryption is after the decode function on transmit (egress). A packet is determined to be a transmit event packet under the following conditions.

##### 9.4.14.2.1 Annex D (IPv4)

1. Transmit time sync is enabled (ts\_tx\_annex\_d\_en is set in the switch Pn\_TS\_Ctl register).
2. One of the sequences below is true.
  - The first packet LTYPE matches 0x0800
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set (PN\_TS\_VLAN\_LTYPE register) and the second packet LTYPE matches 0x0800
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the second packet LTYPE matches 0x0800
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en (PN\_TS\_CTL\_REG) is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the third packet LTYPE matches 0x0800
3. Byte 14 (the byte after the LTYPE) contains 0x45 (IP\_VERSION). Note that the byte numbering assumes that there are no VLANs. The byte number is intended to show the relative order of the bytes. If VLAN(s) are present then the byte numbers push down.
4. Byte 20 contains 0bXXX00000 (5 lower bits zero) and Byte 21 contains 0x00 (fragment offset zero)
5. Byte 22 contains 0x01 (HOP Limit = 1) if the ts\_ttl\_nonzero bit in the switch Pn\_TS\_Ctl\_Ltype2 register is zero, or byte 22 contains any value if ts\_ttl\_nonzero bit is set. Byte 22 is the TTL/HOP field.
6. Byte 23 contains 0x11 (Next Header UDP Fixed).
7. The pn\_ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is zero and bytes 30 through 33 contain:
  - Decimal 224.0.1.129 and the ts\_129 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.1.130 and the ts\_130 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.1.131 and the ts\_131 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.1.132 and the ts\_132 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 224.0.0.107 and the ts\_107 bit in the Pn\_TS\_Ctl\_Ltype2 register is set

Or:

The ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is set and Bytes 30 through 33 contain any values.
8. Bytes 36 and 37 contain:
  - Decimal 0x01 and 0x3f respectively and the ts\_319 bit in the Pn\_TS\_CTL\_Ltype2 register is set, or
  - Decimal 0x01 and 0x40 respectively and the ts\_320 bit in the Pn\_TS\_CTL\_Ltype2 register is set.
9. The PTP message begins in byte 42 (this is offset 0).
10. The packet message type is enabled in ts\_msg\_type\_en field in the Pn\_TS\_Ctl register.
11. The packet was sent by the host (port 0).

#### 9.4.14.2.2 Annex E (IPv6)

1. Transmit annex E time sync is enabled (ts\_tx\_annex\_e\_en is set in the Pn\_TS\_Ctl register).
2. One of the sequences below is true.
  - The first packet LTYPE matches 0x86dd.
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set and the second packet LTYPE matches 0x86dd.
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the second packet LTYPE matches 0x86dd.
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the third packet LTYPE matches 0x86dd.
3. Byte 14 (the byte after the LTYPE) contains 0x6X (IP\_VERSION in most significant nibble).
4. Byte 20 contains 0x11 (UDP Fixed Next Header).
5. Byte 21 contains 0x01 (HOP Limit = 1) if the ts\_ttl\_nonzero bit in the switch Pn\_TS\_Ctl\_Ltype2 register is zero, or byte 21 contains any value if ts\_ttl\_nonzero is set. Byte 21 is the TTL/HOP field.
6. The ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is zero and Bytes 38 through 53 contain:
  - FF0M:0:0:0:0:0:0:0181 and the ts\_129 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0:0182 and the ts\_130 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0:0183 and the ts\_131 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0:0184 and the ts\_132 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - FF0M:0:0:0:0:0:0:006B and the ts\_107 bit in the Pn\_TS\_Ctl\_Ltype2 register is set (all values above are 16-bit hex numbers with M is enabled in the ts\_mcast\_type\_en field in the Pn\_TS\_Ctl2 register).

Or:

The pn\_ts\_uni\_en bit in the Pn\_TS\_Ctl\_Ltype2 register is set and Bytes 38 through 53 contain any value.

7. Bytes 56 and 57 contain (UDP Header in bytes 54 through 61):
  - Decimal 0x01 and 0x3f respectively and the ts\_319 bit in the Pn\_TS\_Ctl\_Ltype2 register is set, or
  - Decimal 0x01 and 0x40 respectively and the ts\_320 bit in the Pn\_TS\_Ctl\_Ltype2 register is set.
8. The PTP message begins in byte 62.
9. The packet message type is enabled in the ts\_msg\_type\_en field in Pn\_TS\_Ctl.
10. The packet was sent by the host (port 0).

#### 9.4.14.2.3 Annex F (IEEE 802.3)

1. Transmit time sync is enabled (ts\_tx\_annex\_f\_en is set in the switch Pn\_TS\_Ctl register).
2. One of the sequences below is true.
  - The first packet LTYPE matches ts\_ltype1. LTYPE 1 should be used when only one time sync LTYPE is to be enabled.
  - The first packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set and the second packet LTYPE matches pn\_ts\_ltype1
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the second packet LTYPE matches pn\_ts\_ltype1
  - The first packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the second packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the third packet LTYPE matches pn\_ts\_ltype1
  - The first packet LTYPE matches ts\_vlan\_ltype1 and ts\_tx\_vlan\_ltype1\_en is set and the second packet LTYPE matches ts\_vlan\_ltype2 and ts\_tx\_vlan\_ltype2\_en is set and the third packet LTYPE matches ts\_ltype2 and ts\_ltype2\_en is set
3. The packet message type is enabled in the ts\_msg\_type\_en field in the Pn\_TS\_Ctl register.



4. The packet was sent by the host (port 0).

The TS\_TX\_MII interface issues a single clock record signal at the beginning of each transmitted packet. If the packet is a time sync event packet then a single clock event signal along with a handle will be issued before the next record signal for the next packet. The event signal will not be issued for packets that did not meet the time sync event criteria in the TS\_TX\_DEC function. If consecutive record indications occur without an interleaving event indication, then the packet associated with the first record was not a time sync event packet. The record signal is a single clock pulse indicating that a transmit packet egress has been detected at the associated port MII interface. The handle value is incremented with each time sync event packet and rolls over to zero after 7. There are 8 possible handle values so there can be a maximum of 8 time sync event packets “in flight” from the TS\_TX\_DEC to the TS\_TX\_MII block at any given time. The handle value increments only on time sync event packets. The TS\_TX\_MII logic is in the transmit wireside clock domain.

#### Note

TS\_TX\_DEC and TS\_TX\_MII are internal interfaces that have no registers. They are included for description purposes only.

### 9.4.15 Rate Limiting (Traffic Shaping) (P802.1Qav/D6.0)

Rate-limit mode is intended to allow some CPPI ingress channels and some Ethernet transmit (switch egress) priorities to be rate-limited. Non rate-limited traffic (bulk traffic) is allowed on lower priority non rate-limited channels and FIFO priorities. Rate-limited traffic must be configured to be sent to rate-limited queues (via packet priority handling). The allocated rates for rate-limited traffic must not be oversubscribed. For example, if port 1 is sending 15% rate limited traffic to port 2 priority 3, and port 0 is also sending 10% rate-limited traffic to port 2 priority 3, then the port 2 priority 3 egress rate must be configured to be 25% plus a percent or two for margin. The switch must be configured to allow some percentage of non rate-limited traffic. Non rate-limited traffic must be configured to be sent to non rate-limited queues. No packets from the host should be dropped, but non rate-limited Ethernet ingress traffic can be dropped. For rate limited priorities, the configured transfer rate includes the committed information rate and the excess information rate. The excess information rate will only be attempted to be sent when there is no packet backlog on every priority that does not have the excess information rate enabled. The committed information rate will be sent regardless of network traffic as long as the configuration is not oversubscribed. The excess information rate will be sent only when network conditions allow.

#### Credit Based Shaper (CBS)

CBS based traffic shaping is supported in the CPSW IP and can be enabled by either configuring it in the **committed information rate** register either in the CPPI port ingress or in the Ethernet port transmit. If the rate limiting is applied at the CPPI port ingress, then the CPDMA makes sure to transmit the data at the limited rate. If it is configured at the Ethernet port end the burst data may be buffered in the Ethernet FIFO before transmitting, care must be taken such that Ethernet FIFO does not overflow when rate limiting is set in Ethernet port.

The relation between CBS parameters and hardware register configuration is as follows:

- idleSlope is essentially the rate limit, i.e.  $\text{idleSlope} = \text{CIR (committed information rate)}$  in Mbps
- portTransmitRate is link speed or port speed which is 100Mbps in AWR2544
- sendSlope is calculated as,  $\text{sendSlope} = \text{idleSlope} - \text{portTransmitRate}$  (this is described in the spec)
- hicredit and locredit : credit is calculated based on the idleSlope and sendSlope, it is maintained internally in the hardware, this is not a configurable parameter to the user.

#### 9.4.15.1 CPPI Port Ingress Rate Limiting

Port 0 FHost operations can be configured to rate limit the packet data for each channel (priority). FHost has 8 priorities for QOS. There is a committed information rate (**P0\_Pri\_CIR**) and an excess information rate for each priority (**P0\_Pri\_EIR**). Rate limiting is enabled for a priority when the committed information rate for the priority is non-zero. The excess information rate for a priority is enabled when the excess information rate for the priority is non-zero. The committed information rate must be non-zero if the excess information rate is configured to be non-zero. That is, there must be a configured non-zero committed information rate for there to

be a configured non-zero excess information rate. Bulk traffic on other non-rate limited priorities does not impact the committed information traffic on a priority. However, bulk traffic on other non-rate limited priorities does impact the excess information rates. No bulk priority will be enabled to send unless there are **tx\_host\_blks\_rem** number of unused blocks remaining in each of the Ethernet port transmit FIFOs. The “blocks remaining check” ensures that bulk traffic from the host will not block rate-limited traffic from the host. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then priorities 7 and 6 should be configured for committed information (and excess information if desired). When any channels are configured to be rate-limited, the ingress priority type must be fixed. Round-robin priority type is not allowed when rate-limiting is configured for any priority. The configured transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second for each priority is controlled by the below equation. If the configured excess information rate is zero, then only the committed information rate is transferred:

$$\text{Priority Transfer rate in Mbit/s} = (((\text{Frequency in MHZ}) * \mathbf{P0\_Pri\_CIR}) / 32768) + ((\text{Frequency in MHZ}) * \mathbf{P0\_PriX\_EIR}) / 32768))$$

Where the frequency is the **VBUSP\_GCLK** frequency (350 for 350Mhz) and priX = pri0 to pri7. For example, 10Mbps on priority 7 would give the below:

$$10\text{Mbps} = \sim ((350 * 936) / 32768), \text{ at } 350\text{Mhz and } \mathbf{p0\_pri7\_cir} \text{ value} = 936 \text{ (no excess information rate)}$$

#### 9.4.15.2 Ethernet Port Transmit Rate Limiting

Ethernet port transmit operations can be configured to rate limit egress data for each egress priority. There is a committed information rate (**Pn\_PriX\_CIR**) and an excess information rate for each priority (**Pn\_PriX\_EIR**). Rate limiting is enabled for a priority when the committed information rate for the priority is non-zero. The excess information rate for a priority is enabled when the excess information rate for the priority is non-zero. The committed information rate must be non-zero if the excess information rate is configured to be non-zero. That is, there must be a configured non-zero committed information rate for there to be a configured non-zero excess information rate. Bulk traffic on other non-rate limited priorities does not impact the committed information traffic on a priority. However, bulk traffic on other non-rate limited priorities does impact the excess information rates. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then priorities 7 and 6 should be configured for committed information (and excess information if desired). The configured transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second that each priority is configured to send is controlled by the below equation. If the excess information rate is disabled then the committed information rate only is transferred:

$$\text{Priority Transfer rate in Mbit/s} = (((\text{Frequency in MHZ}) * \mathbf{Pn\_PriX\_CIR}) / 32768) + ((\text{Frequency in MHZ}) * \mathbf{Pn\_PriX\_EIR}) / 32768))$$

Where the frequency is the **VBUSP\_GCLK** frequency (350 for 350Mhz) and priX = pri0 to pri7. For example, 100Mbps on priority 7 would give the below:

$$100\text{Mbps} = \sim ((350 * 9360) / 32768), \text{ at } 350\text{Mhz and } \mathbf{pn\_pri7\_cir} \text{ value} = 9360, \text{ with no excess information rate.}$$

#### 9.4.16 Transmit Priority Escalation

Bulk (non rate limited) traffic can be escalated in order to preclude starvation of lower bulk traffic priorities from higher bulk traffic priorities. Escalation is configured and enabled in the **PType** register with the **pn\_ptype\_esc** and **esc\_pri\_id\_val** fields. Escalation can be enabled with rate limited priorities. The rate limited priorities are on the upper priorities starting with priority 7 and going down(consecutively), and therefore, the escalated priorities are on the lower bulk priorities.

#### 9.4.17 Enhanced Scheduled Traffic (EST – P802.1Qbv/D2.2)

#### 9.4.17.1 EST Overview

- When enabled and configured, EST allows express queue traffic to be scheduled (placed) on the wire at specific repeatable time intervals.
- EST operates on a repeating time interval generated by the CPTS EST function generator. For example, a 125us repeating time interval can be configured.
- Each Ethernet port has 128 EST fetch commands maximum in the global EST fetch RAM.
- Each 22-bit fetch command consists of a 14-bit fetch count (14 msb's) and an 8-bit priority fetch allow (8 lsb's) that will be applied for the fetch count time in wireside clocks.
- The configured port fetch commands are executed in sequence, beginning at port address zero each time through the time interval beginning at cycle start.
- EST allows non-scheduled express and preempt queue traffic to be cleared from the wire to ensure that the scheduled traffic is transmitted at the proper time (wire clear is performed with zero allow).
- EST can be used with or without preemption. The pn\_mac\_preempt[7:0] value determines whether the priority is enabled on the express or preempt queue. Whether a priority is on the express or preempt queue only effects the wire clear time from an EST operation perspective.
- Software should not move priorities to the preempt queue unless preemption is configured, enabled, and verified - allowing preemption to occur.
- Express packet time stamp events can be enabled to assist software in configuring and timing EST operations.

#### 9.4.17.2 EST Fetch RAM

- The EST fetch RAM is read/write in the CPSW configuration address space.
- Each Ethernet transmit port has 128 locations in the global EST fetch RAM.
  - Ethernet port 1 has EST fetch RAM addresses 0x000-0x1ff.
  - Ethernet port 2 has EST fetch RAM addresses 0x200-0x3ff, and so on.
- One buffer operation – When est\_onebuf is set, the 128 port locations operate as one buffer. The est\_bufact bit in Pn\_FIFO\_Status is the upper address bit of the port's fetch RAM address indicating whether operation is currently in the upper or lower 64 locations of the port's fetch RAM.
- Two buffer operation - When est\_onebuf of PN\_EST\_CONTROL register is cleared there are two 64-location buffers with est\_bufsel selecting the buffer to be used. When the buffer is switched by changing the est\_bufsel value, the actual switch occurs on cycle start. The actual buffer being used is indicated by the est\_bufact bit in Pn\_FIFO\_Status. Software should avoid writing the switched out buffer fetch RAM locations until it detects that the actual switch has occurred.
- The first address location in the port's fetch RAM space (location zero) is read at the beginning of each EST time interval (cycle start). Addresses are then read in ascending order for the duration of the interval. The port's address zero location is then read again at the beginning of the next cycle repeating the time interval packet operations.

#### 9.4.17.3 EST Time Interval

- Each Ethernet port has an EST function (ESTF) generator in the CPTS submodule.
- The EST function generator generates the EST time interval as a configured number of CPTS reference clocks (CPTS\_RCLK).
- The EST function generator rising edge is the cycle start time and the cycle repeats (cycle start occurs) after every time interval.
- The first fetch allow (8 lsb's of the fetch command from the EST fetch RAM) value is at the port's base address zero in the EST fetch RAM and is actually applied 16 wireside clocks after cycle start. The 16 clock delay allows the first fetch value to be fetched from the EST fetch RAM (prefetch time at cycle start).
- Each successive fetch allow is applied for the associated fetch count thereafter. The minimum non-zero fetch count is 16. The minimum value of 16 guarantees that the next fetch value has time to be fetched before the current fetch count is over. There are 64 maximum fetch values when est\_onebuf=0, and 128 maximum fetch values when est\_onebuf=1.
- The next cycle start then causes the fetch to once again start at the port's address zero location.

#### 9.4.17.4 EST Fetch Values

- The 22-bit fetch value is made up of the 14-bit fetch count and the 8-bit fetch allow.
- The fetch time indicates the number of wireside clocks that the fetch allow will be active.
- The fetch count is in Ethernet wireside clocks which is bytes in gigabit mode (pn\_gig=1) and nibbles in 10/100Mbps mode.
- When a fetch allow bit is set, the corresponding priority is enabled to begin packet transmission on an allowed priority subject to rate limiting. There is no requirement that the packet end in the time interval. The actual packet transmission on the wire may carry over into the next fetch count - which is the reason for the wire clear time in a fetch zero allow.
- When a fetch allow bit is cleared, the corresponding priority is not enabled to transmit for the fetch count time. However, if a packet were enabled in a previous fetch allow there the packet could finish in the current time interval.
- A non-zero fetch allow value with a non-zero fetch count causes the fetch allow value to be applied for the fetch count number of wireside clocks (minimum of 16 fetch count).
- A zero fetch count causes the associated fetch allow to be held for the duration of the cycle (until the next cycle start).
- A zero fetch allow with a non-zero fetch count is intended to clear the wire for a scheduled (timed) express packet in the next fetch. A zero fetch allow indicates that no packet can be started for transmission for the associated fetch count. However, packets that were started in the previous interval could still be on the wire. The associated fetch count must be sufficient to guarantee that the wire is cleared given that a packet on an allowed priority in the previous fetch could have been started on the previous clock and that there is hardware latency in the clear time. The timed packet should be sent on a priority that is enabled in the next fetch but disabled in the current zero allow fetch.
- EST Ram Configuration – The simplest EST configuration is for a single express packet on a single priority in the EST time interval. Ram Address 0 will be read at the start of each interval, the timed packet goes out in Ram Address 1 interval, and Ram Address 2 priorities are held for the duration of the cycle in this simple configuration.
  - Ram Address 0 – 0xD0000 (10/100 with 1518 max pkt length in previous allow)
    - This is a zero allow at the beginning of each cycle which clears the wire for the timed express packet in the next fetch. The zero allow time (0xD00 in this case) allows the wire to be cleared for the express packet. Setting this value lower might push out the timed express packet into the next interval. For a 2020 max packet size in the previous allow, this value should be 0x10F400. The number of clocks required for the zero allow is determined by the below equation.
      - Clocks in zero allow = Maximum **Express** packet length in previous allow (times 2 for 10/100) + decimal 292. (If preemption is configured and enabled then this minimum time is 0x100 if only preempt priorities are in the previous allow – which is not a normal case).
  - Ram Address 1 – 0x01080
    - This is a decimal 16 clock allow for the timed express packet on priority 7. The 16 clock allow value permits only a single packet to start in the time interval.
  - Ram Address 2 – 0x0007F
    - This is a zero fetch count that allows priorities 6 down to 0 for the duration of the EST time interval. This is the previous allow for the zero allow in Address 0 above.

#### 9.4.17.5 EST Packet Fill

- Packet fill can (should) be configured and enabled to occur in the fetch count time associated with a fetched zero allow that preceeds a timed express packet. The intention with fill is that a smaller packet on a non-timed priority might be able to be inserted on the wire during the wire clear time which would increase wire utilization. Fill must be configured to ensure that any fill packet does not conflict with the timed express packet allowed in the next fetch. Incorrect configuration might push out in time any express timed packet which indicates that the fill margin needs to be increased.
- Fill Configuration
  - The est\_fill\_margin value in Pn\_EST\_Control should be written with a 0x100 value.

- The `est_preempt_comp` value in `Pn_EST_Control` should be written with a 0x12 value (if IET is to be configured and enabled). This value times eight is the number of wireside clocks required to clear preempt packets off the wire at the end of a zero allow.
- The `est_fill_en` bit in `Pn_EST_Control` should be set.

#### 9.4.17.6 EST Time Stamp

- EST can be configured to generate CPTS timestamp events for selected express traffic.
- EST timestamp events use the CPTS host event type (`event_type=7`). EST timestamps will not override host sent timestamps for packets that were sent from the host with an enabled host timestamp.
- EST Events (host events `EVENT_n_REG`) contain the below information:
  - Time Stamp of the selected express packet.
  - The event `port_number` indicates the transmit port number.
  - The event `event_type` is decimal 7 (host event).
  - The event `message_type` indicates the packet transmit hardware switch priority.
  - The event `sequence_id` upper nibble indicates the packet receive port number.
  - The event `sequence_id` lower byte indicates the sequence number of the express packet in numerical order. The first event is event one, the second is event two and so on. The sequence id rolls over to zero after 0xff (8-bits).
  - The event domain is the value from the `est_ts_domain[8:0]` register.
- When `est_ts_en` is set, timestamp events will be generated on selected express traffic.
- When `est_ts_first` is also set, events will be generated only on the first express packet in each time interval. If `est_ts_onepri` is also set then the event will only be on the first `est_ts_pri` express packet in the time interval. If `est_ts_onepri` is clear then the event will be generated on the first express packet in the time interval on any priority.
- When `est_ts_first` is clear, events will be generated on every express packet. If `est_ts_onepri` is set then the event will be generated on every `est_ts_pri` express packet. If `est_ts_onepri` is clear then event will be generated on every express packet on any priority.

#### 9.4.17.7 EST Packets Per Priority – (N = 2 only)

With a MAC configuration ( $N = 2$ ), the number of packets allowed in a transmit FIFO priority can be selected by writing a non-zero value to `p0_rx_pkts_pri[7:0]`. The port 0 receive gap should then be enabled by setting the corresponding priority `rx_gap_en[7:0]`. The receive gap allows a packet to land in the transmit FIFO before another packet is allowed in which guarantees that only the selected number (max) of packets is allowed in on the specified priority. If the receive gap is not enabled, then there might be one or two more packets allowed in on the priority than the `p0_rx_pkt_pri[7:0]` value has selected.

#### 9.4.18 DSCP

The ALE can map DSCP field to priority prior to port trunking hashing and policing/classification matching. When enabled the DSCP is mapped via 64 priority entries such that any DSCP value can be mapped to any of the eight priorities. When a packet is received without a VLAN priority this remapped priority can be used instead of the default Port VLAN priority field. See the Switch Port configuration for the registers describing DSCP mapping.

#### 9.4.19 Packet Priority Handling

There are three priorities used inside the CPSW - the packet priority, the header packet priority, and the switch priority. The packet priority is the determined priority of the ingress packet. The header packet priority is used as the outgoing VLAN priority if the packet is egressing from the switch with a VLAN tag. The switch priority determines which of the eight FIFO priority queues the packet uses during egress.

The `vlan_ltype_sel` value below is selected by the `s_cn_switch` bit in the `CPSW_Control` register and is either the `vlan_ltype_inner` (0x8100 default) or `vlan_ltype_outer` (0x88a8 default) value.

##### 9.4.19.1 Ethernet Port Ingress

Ethernet ingress packets have an ingress packet priority of 0 to 7 (with 7 being the highest priority). The packet priority is determined as follows:



1. If the first packet LTYPE = vlan\_ltype\_sel then the ingress packet priority is the packet priority (VLAN tagged and priority tagged packets).
2. Else if the first packet LTYPE = 0x0800 and byte 14 (following the LTYPE) is equal to 0x4X, and dscp\_ipv4\_en is set in Pn\_Control, then the ingress packet priority is the 6-bit TOS field in byte 15 (upper 6-bits) mapped through the port's DSCP priority mapping registers (IPv4 packet).
3. Else if the first packet LTYPE = 0x86dd and the most significant nibble of byte 14 (following the LTYPE) is equal to 0x6, and dscp\_ipv6\_en is set in Pn\_Control, then the ingress packet priority is the 6-bit priority (in the 6-bits following the upper nibble 0x6) mapped through the port's DSCP priority mapping registers (IPv6 packet).
4. Else the ingress packet priority is the source (ingress) port priority taken from the port's Pn\_Port\_VLAN register.

The packet priority is mapped through the ingress port's associated "packet priority to header packet priority mapping register" (**pn\_rx\_pri\_map**) to obtain the header packet priority (**hdr\_pkt\_pri**). The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress. The header packet priority is mapped at each destination FIFO through the **pn\_tx\_pri\_map** register (header priority to switch priority mapping register) to obtain the hardware switch priority (hardware queue 0 through 7).

#### 9.4.19.2 CPPI Port FHost (Ingress)

FHost packets have a packet priority (0 to 7 with 7 being the highest priority). The FHost packet priority is determined as follows:

1. If the first packet LTYPE = vlan\_ltype\_sel then the FHost packet priority is the packet priority (VLAN tagged and priority tagged packets).
2. Else if the first packet LTYPE = 0x0800 and byte 14 (following the LTYPE) is equal to 0x4X, and p0\_dscp\_ipv4\_en is set in P0\_Control, then the FHost packet priority is the 6-bit TOS field in byte 15 (upper 6-bits) mapped through the port's DSCP priority mapping registers (IPv4 packet).
3. Else if the first packet LTYPE = 0x86dd and the most significant nibble of byte 14 (following the LTYPE) is equal to 0x6, and dscp\_ipv6\_en is set in P0\_Control, then the FHost packet priority is the 6-bit priority (in the 6-bits following the upper nibble 0x6) mapped through the port's DSCP priority mapping registers (IPv6 packet).
4. Else the FHost packet priority is the source port priority taken from P0\_Port\_VLAN.

The ingress packet priority is mapped through the port's associated "packet priority to header packet priority mapping register" (**p0\_rx\_pri\_map**) to obtain the header packet priority. The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress.

For CPPI FHost packets, the destination port hardware switch priority is the below selected value remapped through **p0\_rx\_pri\_map**:

1. If the FHost packet is priority tagged or vlan tagged:
  - If rx\_remap\_vlan is clear then the destination hardware switch priority is the CPPI FHost channel number.
  - If rx\_remap\_vlan is set then the destination hardware switch priority is the packet priority value. Port transmit remapping (Pn\_Tx\_Pri\_Map should remain the default value) is not compatible with this bit being set, but remapping can be configured on port 0 FHost. If N=2 (two port MAC) remapping should be done only on ingress for Ethernet and CPPI.
2. Else if the ingress packet has the first packet LTYPE = 0x0800 and byte 14 (following the LTYPE) is equal to 0x4X, and **dscp\_ipv4\_en** is set in **P0\_Control**:
  - If rx\_remap\_dscp\_v4 is clear then the destination hardware switch priority is the CPPI ingress priority.
  - If rx\_remap\_dscp\_v4 is set then the destination hardware switch priority is the 6-bit TOS field in byte 15 (upper 6-bits) mapped through the port's DSCP priority mapping registers (IPv4 packet). Port 1 transmit remapping (Pn\_Tx\_Pri\_Map should remain the default value) is not compatible with this bit being set, but remapping can be configured on port 0 ingress. If N=2 (two port MAC) remapping should be done only on ingress for Ethernet and CPPI.

3. Else if the ingress packet has the first packet LTYPE = 0x86dd and the most significant nibble of byte 14 (following the LTYPE) is equal to 0x6, and **dscp\_ipv6\_en** is set in **CPPI\_P0\_Control**:
  - If **rx\_remap\_dscp\_v6** is clear then the destination hardware switch priority is the CPPI ingress priority.
  - If **rx\_remap\_dscp\_v6** is set then the destination hardware switch priority is the 6-bit priority (in the 6-bits following the upper nibble 0x6) mapped through the port's DSCP priority mapping registers (IPv6 packet). Port 1 transmit remapping (**Pn\_Tx\_Pri\_Map** should remain the default value) is not compatible with this bit being set, but remapping can be configured on port 0 ingress. If N=2 (two port MAC) remapping should be done only on ingress for Ethernet and CPPI.
4. Else the ingress packet is non-tagged and the destination hardware switch priority is the CPPI ingress channel number.

#### 9.4.19.3 CPPI Port THost (Egress)

If the **thost\_ch\_override** bit in **CPDMA\_Control** is clear then the CPDMA packet THost channel number is the port 0 hardware switch priority. If **thost\_ch\_override** is set, then for packets with a classification match the THost channel number is the lower three bits of the 6-bit address lookup engine classification match value (**threadval[2:0]** in ALE register **THREADMAPVAL**). The **flow** value in the VLAN encapsulation word is all 6 bits of the **threadval** for classifier matches regardless of the setting of **thost\_ch\_override** if the encapsulation word is transferred.

#### 9.4.20 Packet CRC Handling

Every cpsw ingress packet on all ports is checked for CRC correctness. Each packet is then given an internally generated Castagnoli CRC for transport through the cpsw to egress port(s). The internally generated Castagnoli CRC protects the packet from end to end through the cpsw. Ingress Packets with CRC errors are handled as indicated below.

##### 9.4.20.1 Ethernet Port Ingress

Ethernet ports check each ingress packet CRC for correctness in all modes/speeds. The port can check for either Ethernet CRC for correctness as determined by the **crc\_type** bit in the **Mac\_Control** register. Ethernet packets received with a CRC error are dropped at the receive port unless the receive port **rx\_cef\_en** bit is set. Error packets are sent only to the host port. For packets with Ethernet ingress errors sent to the host, the error is indicated in the host egress buffer descriptor and the original packet CRC bytes are kept for CPDMA THost.

##### 9.4.20.2 Ethernet Port Egress

Ethernet ports transmit egress packets with the CRC type selected by the **crc\_type** bit in the **Pn\_Mac\_Control** register regardless of the packet's ingress CRC type. On Ethernet egress, after passing through the switch, the internally generated Castagnoli CRC is checked for correctness and if correct the packet is output with the generated selected output CRC type. If the internally generated CRC is incorrect, due either to a bit flip in a memory (or logic) or an error CRC passed in on host ingress, then the generated egress CRC type is used with at least a single byte of the internally generated Castagnoli CRC inverted to indicate the error. If the packet length including CRC is divisible by 4 then all 4 CRC bytes will be inverted on error. If there are three bytes remainder after dividing the packet length by 4 then three bytes will be inverted (and so on down to one byte remainder).

##### 9.4.20.3 CPPI Port FHost

CPPI host port FHost packets can be passed in with or without a CRC. The FHost packet CRC type is indicated in the buffer descriptor word **crc\_type** bit and can be Ethernet. The **p0\_rx\_pass\_crc\_err** bit in the **CPSW\_Control** register determines if FHost packets with CRC errors are passed or dropped. Passed packets with CRC errors will be transmitted on Ethernet egress with a CRC error.

##### 9.4.20.4 CPPI Port Egress (THost)

The **p0\_tx\_crc\_remove** bit in the **CPSW\_Control** register determines if all CPPI THost packets have a CRC included or not. THost packets not filtered on Ethernet ingress due to **pn\_rx\_cef\_en** have the packet error CRC included (not replaced by the egress CRC type) if the CRC is not removed on egress. The error is indicated in the buffer descriptor. CPPI THost packets that detected a CRC error on the internally generated Castagnoli CRC, due to a bit flip in logic or memory, will indicate the error with the **drop** bit set in the buffer descriptor.

### 9.4.21 Ethernet FIFO

Each transmit packet FIFO contains eight logical transmit queues (priority 0 through 7 with 7 the highest priority) regardless of the memory configuration. The receive logical FIFO is combined with the transmit logical FIFO into the same RAM instance. The receive FIFO is used for single packet reception and error detection and flow control runout. Packets are queued on transmit. Ethernet FIFO size is 20 KBytes.

### 9.4.22 Ethernet Receive Flow Control

When enabled and triggered, receive flow control is initiated to limit the Mac from further frame reception. Half-duplex mode receive flow control is collision based while full duplex mode issues 802.3X or Priority Based Flow Control (PFC) pause frames. In all cases, receive flow control prevents frame reception by issuing the flow control appropriate for the current mode of operation. Receive flow control is enabled by the **rx\_flow\_en** bit in the **Pn\_Mac\_Control** register, and **rx\_flow\_pri[7:0]** in **Pn\_Pri\_Ctl**. For 10/100 modes of operation, collision or IEEE 802.3X flow control is determined via the **fullduplex** bit in the **Pn\_Mac\_Control** register. The **fullduplex** bit must be set for Priority Based Flow Control.

#### 9.4.22.1 MII (10/100) Collision Based Receive Buffer Flow Control

Collision-based receive buffer flow control provides a means of preventing frame reception when the port is operating in half-duplex mode (**fullduplex** is cleared in **Pn\_Mac\_Control**). When receive flow control is enabled and triggered, the port will generate collisions for received frames. The jam sequence transmitted will be the twelve byte sequence C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3 (hex). The jam sequence will begin no later than approximately as the source address starts to be received. Note that these forced collisions will not be limited to a maximum of 16 consecutive collisions, and are independent of the normal back-off algorithm. Receive flow control does not depend on the value of the incoming frame destination address. A collision will be generated for any incoming packet, regardless of the destination address.

#### 9.4.22.2 IEEE 802.3X (10/100/1G/10G) Receive Flow Control

IEEE 802.3x based receive flow control provides a means of preventing frame reception when the port is operating in full-duplex mode (**fullduplex** is set in **Pn\_Mac\_Control**). When receive flow control is enabled and triggered, the port will transmit a pause frame to request that the sending station stop transmitting for the period indicated within the transmitted pause frame.

The Mac will transmit a pause frame to the reserved multicast address at the first available opportunity (immediately if currently idle, or following the completion of the frame currently being transmitted). The pause frame will contain the maximum possible value for the pause time (0xFFFF). The MAC will count the receive pause frame time (decrements 0xFF00 down to zero) and retransmit an outgoing pause frame if the count reaches zero. When the flow control request is removed, the MAC will transmit a pause frame with a zero pause time to cancel the pause request.

Note that transmitted pause frames are only a request to the other end station to stop transmitting. Frames that are received during the pause interval will be received normally (provided the Rx FIFO is not full at which time the receive FIFO will overrun and **Rx\_Top\_Of\_FIFO\_Drop** will increment).

Pause frames will be transmitted if enabled and triggered regardless of whether or not the port is observing the pause time period from an incoming pause frame.

The Mac will transmit pause frames as described below:

- The 48-bit reserved multicast destination address 01.80.C2.00.00.01.
- The 48-bit source address – Pn\_SA(47:0) (PN\_SA\_L, PN\_SA\_H registers).
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause time value FF.FF. A pause-quantum is 512 bit-times. Pause frames sent to cancel a pause request will have a pause time value of 00.00.
- Zero padding to 64-byte packet length (The MAC will transmit only 64 byte pause frames).
- The 32-bit frame-check sequence (CRC word).



All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

If **rx\_flow\_en** is cleared to zero while the pause time is nonzero, then the pause time will be cleared to zero and a zero count pause frame will be sent.

#### 9.4.22.2.1 Flow Control Trigger

Receive flow control is triggered (when enabled), when the number of words in the receive FIFO is greater than or equal to **rx\_flow\_thresh**. The flow control packet runout is then contained in the remainder of the receive FIFO.

#### 9.4.22.3 Ethernet Transmit Flow Control

##### 9.4.22.3.1 IEEE 802.3X (10/100/1G/10G) Based Transmit Flow Control

Incoming pause frames are acted upon, when enabled, to prevent the Mac from transmitting any further frames. Incoming pause frames are only acted upon when the **fullduplex** and **tx\_flow\_en** bits in the **Pn\_Mac\_Control** register are set. Pause frames are not acted upon in half-duplex mode. Pause frame action will be taken if enabled, but normally the frame will be filtered and not transferred to memory. MAC control frames will be transferred to memory if the **rx\_cmf\_en** (Copy MAC Frames) bit in the **Pn\_Mac\_Control** register is set. The **tx\_flow\_en** and **fullduplex** bits effect whether or not MAC control frames are acted upon, but they have no effect upon whether or not MAC control frames are transferred to memory or filtered.

Pause frames are a subset of MAC Control Frames with an opcode field=0x0001. Incoming pause frames will only be acted upon by the port if:

- **tx\_flow\_en** is set in **Pn\_Mac\_Control**, and
- the frame's length is 64 to **rx\_maxlen** bytes inclusive, and
- the frame contains no crc error or align/code errors.

The pause time value from valid frames will be extracted from the two bytes following the opcode. The pause time will be loaded into the port's transmit pause timer and the transmit pause time period will begin.

If a valid pause frame is received during the transmit pause time period of a previous transmit pause frame then:

- if the destination address is not equal to the reserved multicast address or any enabled or disabled unicast address, then the transmit pause timer will immediately expire, or
- if the new pause time value is zero then the transmit pause timer will immediately expire, else
- the port transmit pause timer will immediately be set to the new pause frame pause time value. (Any remaining pause time from the previous pause frame will be discarded).

If **tx\_flow\_en** in **Pn\_Mac\_Control** is cleared, then the pause-timer will immediately expire.

The port will not start the transmission of a new data frame any sooner than 512-bit times after a pause frame with a non-zero pause time has finished being received (**MRXDV** going inactive). No transmission will begin until the pause timer has expired (the port may transmit pause frames in order to initiate outgoing flow control). Any frame already in transmission when a pause frame is received will be completed and unaffected.

Incoming pause frames consist of the below:

- A 48-bit destination address equal to:
  - The reserved multicast destination address 01.80.C2.00.00.01, or
  - The **Enet\_Pn\_SA[47:0]** input mac source address.
- The 48-bit source address of the transmitting device.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause\_time. A pause-quantum is 512 bit-times.
- Padding to 64-byte packet length.
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

The padding is required to make up the frame to a minimum of 64 bytes. The standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid pause frames. The MAC will recognize any pause frame between 64 bytes and **rx\_maxlen** bytes in length.

### 9.4.23 Energy Efficient Ethernet Support (802.3az)

Energy Efficient Ethernet (EEE) allows the external clock controller to turn off (in a glitch-less manner) the module input clock (**CLK**) during inactive periods as determined by network and host traffic. The module can then be awakened by host queued transmit packet(s) or by a port's external Ethernet PHY. The module EEE clock stop interface is used by the external controller to control module EEE operations. EEE is supported in MII/GMII/RGMII but not RMII. The chosen PHY must also support EEE. EEE operations are configured as shown below:

1. The 12-bit EEE clock pre-scale value is written to the switch **EEE\_Prescale** register. The pre-scaler is used to clock all EEE related counters.
2. The port Idle to LPI count values (**Pn\_Idle2LPI**) are written with the desired values.
3. The port LPI to Wake count values (**Pn\_LPI2Wake**) are written with the desired values.
4. The **eee\_en** bit is set in the switch **CPSW\_Control** register.

Energy Efficient Ethernet operation can begin after configuration. The host allows the module to enter a low power state by asserting the **CLKSTOP\_REQ** signal. There are no requirements on host queues or traffic in order for the host to assert or de-assert **CLKSTOP\_REQ**.

Each Ethernet port has a transmit and a receive LPI (low power indicate) state. The receive LPI state is entered when the port's corresponding PHY indicates the LPI state via the MII/RGMII interface. The PHY indicates LPI by asserting **GMII\_MRXER\_SEL** with a **GMII\_MRXD[7:0]** value of 0x01 while **GMII\_MRXDV** is deasserted (inter-packet gap). The Ethernet transmit port indicates LPI after the **PX\_Idle2LPI** value has been counted (the transmit port has gone idle for the configured amount of time). If another packet is received for transmit during the count then the count is restarted. When the transmit port has been idle for the Idle to LPI time, the transmit port enters the LPI state and indicates LPI to the associated PHY. The LPI is indicated to the external PHY by an asserted **GMII\_MTXER** with a **GMII\_MTXD[7:0]** value of 0x01. while **GMII\_MTXEN** is deasserted (inter-packet gap). The CPPI LPI state includes transmit and receive. The CPPI LPI state is entered when the CPPI transmit and receive interfaces have both been idle for the Idle to LPI time (**P0\_Idle2LPI**). The Idle to LPI time value for all ports must be large relative to the switch latency to ensure that the count is not able to complete between successive packets.

When all transmit and receive ports are in the LPI state (CPSW LPI state), the **CLKSTOP\_ACK** signal is asserted, and the external clock controller is allowed to stop the input module clock (**CLK**). The clock must be stopped (and re-started) in a glitch-less manner. When **CLKSTOP\_ACK** is asserted, the clock may be turned on and off as desired by the host. The host is allowed to restart the clock, perform slave read/write operations to the module memory address space, and then turn off the clock again while **CLKSTOP\_ACK** is asserted. The clock must be restarted for two clock periods before the assertion of **SLV\_VBUSB\_REQ** and must remain asserted for five clock periods after the de-assertion of **SLV\_VBUSB\_REQ**.

The host can remove and disable from re-entering the CPSW LPI state by restarting the module clock (in a glitch-less manner) and then de-asserting **CLKSTOP\_REQ**. There must be at least one rising edge of the clock before **CLKSTOP\_REQ** is de-asserted. The module **CLKSTOP\_ACK** output signal will be deasserted on the clock after the de-assertion of **CLKSTOP\_REQ**. The host may queue CPPI receive packets at any time without regard to the module LPI state. The Host must deassert **CLKSTOP\_REQ** on wakeup for a minimum of two clock periods. If **CLKSTOP\_REQ** is deasserted for less than 5 clock periods for a wakeup event from the host to a particular Ethernet port (or visa versa), then the wakeup event will not cause the other Ethernet ports to awaken.

The external Ethernet PHY's can also wakeup the external clock controller by removing the Ethernet receive LPI indication. If the module is in the CPSW Idle state with **CLKSTOP\_ACK** asserted and the receive LPI indication

is removed, the **CLKSTOP\_WAKEUP** signal will be asynchronously asserted. On wakeup, the external clock controller must restart the clock and de-assert the **CLKSTOP\_REQ** signal. The **CLKSTOP\_WAKEUP** signal will be synchronously deasserted with **CLKSTOP\_ACK**. There must be at least one positive edge of the clock before the **CLKSTOP\_REQ** signal is deasserted. Upon the de-assertion of **CLKSTOP\_REQ**, the Ethernet ports will count the **Pn\_LPI2Wake** time for each port at which time the port is available for transmit.

#### 9.4.24 Switch Latency

When the CPSW is configured as a store and forward switch, the switch latency is defined as the amount of time between the end of packet reception of the received packet to the start of the output packet transmit. The store and forward latency is shown in the below table:

Mode	Latency
Gig (1000)	880ns
100	1.3us
10	6.5us

#### 9.4.25 Emulation Control

The emulation control input (**EMUSUSP**), emulation control register, and the submodule emulation control registers allow CPSW operation to be completely or partially suspended. The CPDMA Host port and each Ethernet port has associated emulation control registers. The submodule emulation control registers must be accessed to facilitate CPSW emulation control. The CPSW module enters the emulation suspend state if the switch level emulation control register and all submodules are configured for emulation suspend and the emulation suspend input is asserted. A partial emulation suspend state is entered if one or more submodules is configured for emulation suspend and the emulation suspend input is asserted. Emulation suspend occurs at packet boundaries. The emulation control feature is implemented for compatibility with other peripherals. The following table shows the operation of the emulation control input and register bits:

EMUSUSP	soft	free	Description
0	X	X	Normal Operation
1	0	0	Normal Operation
1	1	0	Emulation Suspend
1	X	1	Normal Operation

#### Note

Enable Suspend control (EMUSUSP) by MSS\_CTRL: MSS\_DBG\_ACK\_CTL0[26:24]

#### 9.4.26 Software IDLE

The software idle register bits enable switch operation to be completely or partially suspended by software control. Each Ethernet port may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled port until the idle command is removed. The CPSW software idle inhibits packets from starting to be unloaded from each port switch FIFO, but packets already in process are unaffected.

#### 9.4.27 Common Platform Time Sync (CPTS)

The Common Platform Time Sync (CPTS) module is used to facilitate host control of time sync operations. It enables compliance with the IEEE 1588 standard for a precision clock synchronization protocol.

Main features of CPTS module are:

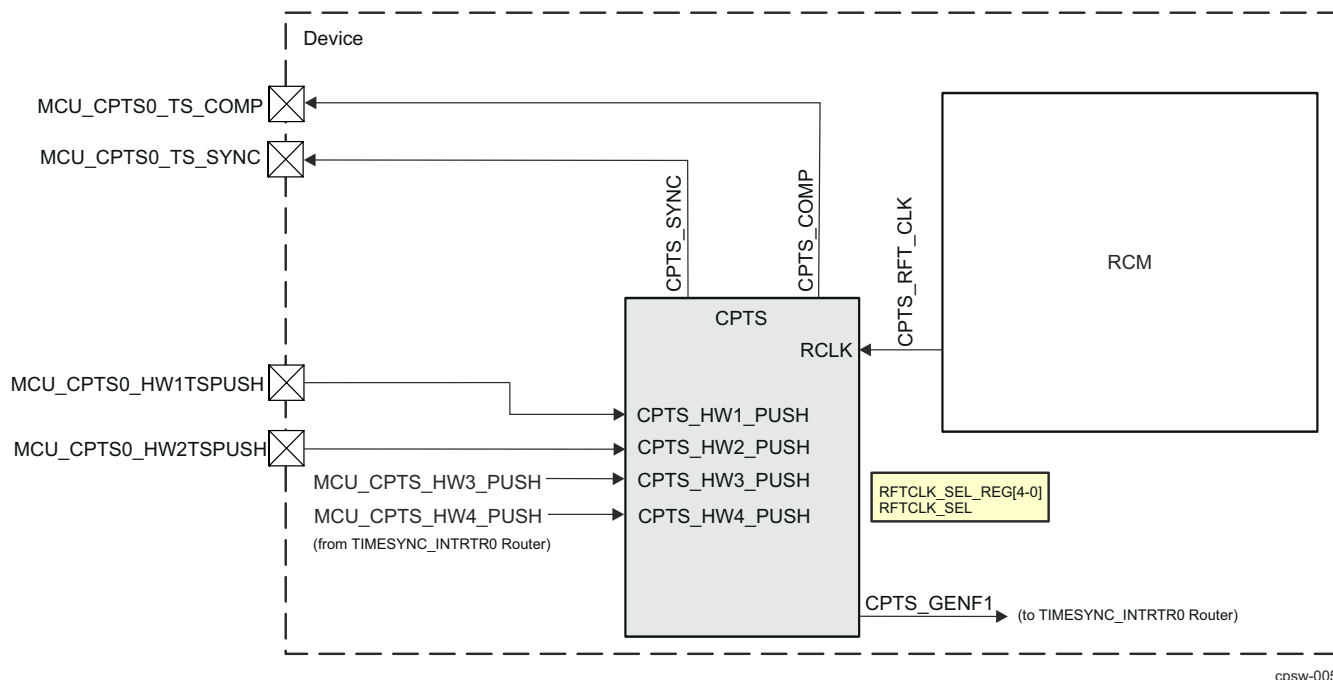
- Supports the selection of up to 32 external clock sources.
- Software control of time sync events via interrupt or polling.
- Supports 8 hardware timestamp push inputs.

- Supports timestamp counter compare output (TS\_COMP).
- Supports timestamp counter bit output (TS\_SYNC).
- Supports a configurable number of timestamp Generator bit outputs (TS\_GENFn).
- Supports Ethernet Enhanced Scheduled Traffic Operations (TS\_ESTFn).
- 64-bit timestamp mode with PPM and nudge adjustment.

#### 9.4.27.1 CPTS Integration

This section describes CPTS module integration in the device, including information about clocks, resets, and hardware requests.

Figure 9-7 shows CPTS integration in the device.



**Figure 9-7. CPTS Integration**

CPTS IEEE 1588 clock (RCLK) is selected through the CPSW\_CPTS\_RFTCLK\_SEL\_REG register.

#### Note

For more information about CPTS clocks and resets, see [Table 9-3](#) in [Section 9.3 CPSW Integration](#).

#### 9.4.27.2 CPTS Architecture

Figure 9-8 shows the architecture of the CPTS module inside the CPSW Ethernet Subsystem. Time stamp values for every packet transmitted or received on either port of the CPSW are recorded. At the same time, each packet is decoded to determine if it is a valid time sync event. If so, an event is loaded into the Event FIFO for processing containing the recorded time stamp value when the packet was transmitted or received.

In addition, both hardware (HWx\_TS\_PUSH) and software (TS\_PUSH) can be used to read the current time stamp value through the Event FIFO. The reference clock used for the time stamp (CPTS\_RFT\_CLK) can be derived from several sources.



frequency possible should be used as allowed by the technology. Table 9-22 shows the RFTCLK\_SEL and TS\_ADD\_VAL\_REG values for 1ns operations.

**Table 9-22. ADD\_VAL feature**

RFTCLK_SEL (MHz)	TS_ADD_VAL_REG[2:0]
1 GHz	0
500 MHz	1
333.33 MHz	2
250 MHz	3
200 MHz	4
166.66 MHz	5
142.85714 MHz	6
125 MHz	7

#### 9.4.27.6 64-Bit Timestamp Nudge

The 64-bit **time\_stamp** value can be adjusted by writing the **ts\_nudge\_val[7:0]** register value which is a 2's complement value. A value of 0xff will subtract 1 **RCLK** from the next incremented **time\_stamp[63:0]** value. A nudge value of 0x01 will add 1 **RCLK** to the next incremented **time\_stamp[63:0]** value. For example, if the current **time\_stamp** value is 0x0f06, and **add\_val[2:0]=3**, the next incremented timestamp value would be 0x0f0a without a nudge and 0x0f0a +/- **tx\_nudge\_val[7:0]** with a nudge. The **ts\_nudge** value is cleared to zero when the nudge has occurred.

#### 9.4.27.7 64-bit Timestamp PPM

The 64-bit **time\_stamp** can be adjusted by parts per million or by parts per hour. Writing a non-zero value to the **ts\_ppm[41:0]** (PPM\_High & PPM\_Low) value enables PPM operations. The adjustment is up or down depending on the **ppm\_dir** bit (TS\_control\_Reg). The **time\_stamp** value is increased by the PPM value when **ppm\_dir** is cleared and decreased by the PPM value when **ppm\_dir** is set.

##### Parts Per Million example:

To adjust for 100 parts per million the configured value for **ts\_ppm[41:0]** is:

$$1,000,000/100 = \text{decimal } 10,000$$

##### Parts Per Hour example:

To adjust for 1 part per hour at 1 Ghz **RCLK** the configured value for **ts\_ppm[41:0]** is: (1,000,000,000hz/1pph) \* (3600 seconds/hour) = hex 34630B8A000

#### 9.4.27.8 Event FIFO

The event FIFO contains at least as many locations as two times the number of ports plus 6 locations. Software must service the event FIFO in a timely manner which prevents event FIFO overrun. No overrun indication will be given.

#### 9.4.27.9 Timestamp Compare Output

The **TS\_COMP** function is a software oriented feature that is intended to be replaced going forward by the hardware oriented GENF function. **TS\_COMP** is not compatible with timestamp PPM or a non-zero **TS\_ADD\_VAL** value.

##### 9.4.27.9.1 Non-Toggle Mode: 32-bit

The **TS\_COMP** output is asserted for **ts\_comp\_length[31:0]** **RCLK** periods when the **time\_stamp[31:0]** (**EVENT\_0\_REG**) value compares with the **ts\_comp\_val[31:0]** and the length value is non-zero. The **TS\_COMP** rising edge occurs three **RCLK** periods after the values compare. A timestamp compare event is pushed into the event FIFO when **TS\_COMP** is asserted. The polarity of the **TS\_COMP** output is determined by the **ts\_comp\_polarity** bit. The output is asserted low when the polarity bit is low.

#### 9.4.27.9.2 Non-Toggle Mode: 64-bit

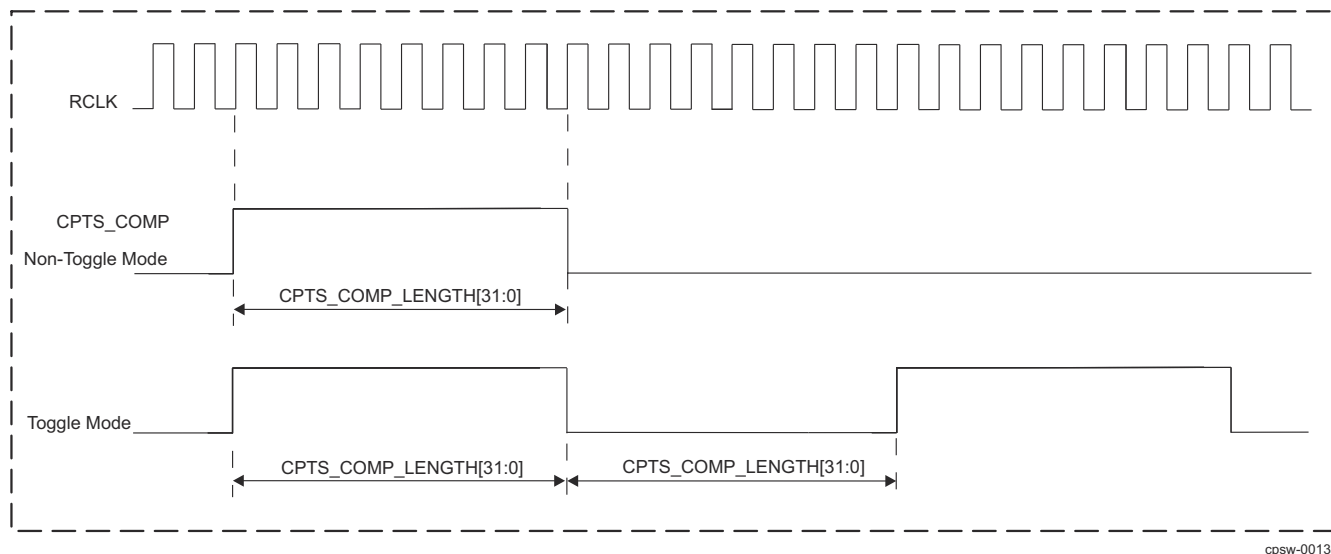
64-bit mode operation is identical to 32-bit mode except that all 64-bits of the **timestamp[63:0]** are used (Event\_0\_reg and Event\_3\_reg) instead of only the lower 32-bits. The 64-bit timestamp cannot be allowed to rollover.

#### 9.4.27.9.3 Toggle Mode: 32-bit

The **TS\_COMP** output is asserted for **ts\_comp\_len[31:0]** RCLK periods when the **time\_stamp[31:0]** value compares with the **ts\_comp\_low\_val[31:0]** and the length value is non-zero. The **TS\_COMP** toggles thereafter on **ts\_comp\_low\_len[31:0]** RCLK periods. The length high or low can be adjusted by writing the **TS\_COMP\_NUDGE[7:0]** register value which is a 2's complement value. A value of 0xff will subtract 1 RCLK from the **ts\_comp\_length[31:0]** value. A value of 0x01 will add 1 RCLK to the **ts\_comp\_length[31:0]** value. Only a single high or low time is adjusted (nudged) and the **ts\_comp\_nudge** value is cleared to zero when the nudge has occurred. The **TS\_COMP** output is asserted low when the **TS\_Comp\_Polarity** bit is low. No compare events and no **CPTS\_EVNT** interrupts are generated in toggle mode. The **ts\_comp\_tog** bit must be set for toggle mode, and must be set before writing a non-zero value to **ts\_comp\_length[31:0]**.

#### 9.4.27.9.4 Toggle Mode: 64-bit

64-bit mode operation is identical to 32-bit mode except that all 64-bits of the **TIMESTAMP** are used (**EVENT\_0\_REG** and **EVENT\_3\_REG**). In 32-bit mode only the lower 32-bits (**EVENT\_0\_REG**) are used.



**Figure 9-9. CPTS\_COMP Output in Toggle and Non-Toggle Mode**

#### 9.4.27.10 Timestamp Sync Output

The **CPTS\_SYNC** output is a selected bit of the [31:0] **TIME\_STAMP** counter value. One of bits 17-31 can be selected in **CPTS\_CONTROL\_REG[31-28]** **TS\_SYNC\_SEL**. The **CPTS\_SYNC** output is disabled when **CPTS\_CONTROL\_REG[31-28]** **TS\_SYNC\_SEL** is zero.

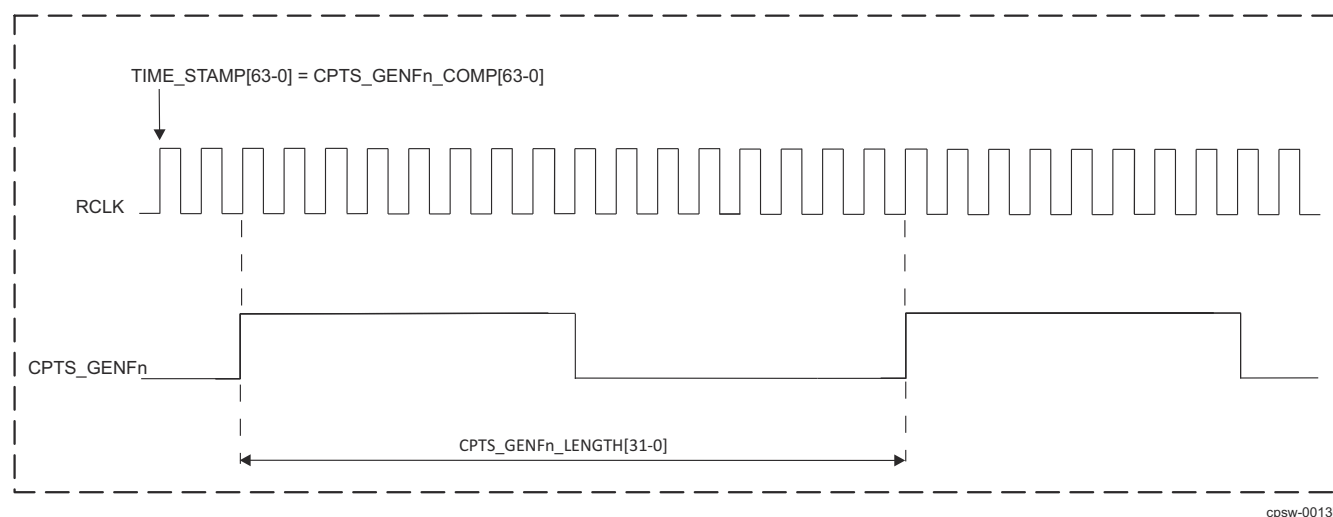
If the selected counter bit is 1 at the time when **TS\_SYNC\_SEL** value is written then a rising edge will not occur on the **CPTS\_SYNC** output. A rising edge will occur on the **CPTS\_SYNC** output upon the next transition to 1 of the selected counter bit. The **TS\_SYNC\_SEL** value must be written to zero before changing to a different non-zero value. No events are generated due to the **CPTS\_SYNC** operation. The **CPTS\_SYNC** output is two **CPTS\_RFT\_CLK** periods after the actual count value.



### 9.4.27.11 Timestamp GENFn Output

The CPTS\_GENFn outputs have a programmable cycle (frequency) with a PPM feature and software nudge feature. The CPTS\_GENFn output cycle is CPSW\_GENF0\_LENGTH\_REG\_I[31-0] CPTS\_RFT\_CLK periods (which is different than CPTS\_COMP operation). [Figure 9-10](#) represents the CPTS\_GENFn output signal.

The CPTS\_GENFn output cycle is CPSW\_GENF0\_LENGTH\_REG\_I[31-0] CPTS\_RFT\_CLK periods beginning when the 64-bit TIME\_STAMP value compares with the 64-bit GENFn\_COMP value (CPSW\_GENF0\_COMP\_LOW\_REG\_I and CPSW\_GENF0\_COMP\_HIGH\_REG\_I registers) and the length value is non-zero. The CPTS\_GENFn output cycle repeats thereafter every CPSW\_GENF0\_LENGTH\_REG\_I[31-0] CPTS\_RFT\_CLK periods. The upper 32-bit word should be written first for 64-bit values. The length should be zero while the comparison value and other configuration parameters are being configured. The length should be written non-zero to enable operations last. The first cycle after comparison is active high when the CPSW\_CPTS\_CONTROL\_REG[2] TS\_COMP\_POLARITY bit is low. No compare events and no CPTS\_EVNT interrupts are generated.



**Figure 9-10. CPTS\_GENFn Output Signal Diagram**

#### 9.4.27.11.1 GENFn Nudge

The cycle length can be adjusted by writing the CPSW\_CPTS\_TS\_COMP\_NUDGE\_REG[7-0] NUDGE register value which is a two's complement value. A value of FFh will subtract 1 CPTS\_RFT\_CLK from the CPSW\_GENF0\_LENGTH\_REG\_I[31-0] value. A value of 1h will add 1 CPTS\_RFT\_CLK to the CPSW0\_ESTF1\_LENGTH\_REG\_I[23-0] value. The CPSW\_CPTS\_TS\_COMP\_NUDGE\_REG[7-0] NUDGE value is cleared to zero when the nudge has occurred.

#### 9.4.27.11.2 GENFn PPM

The CPTS\_GENFn output cycle can be adjusted by parts per million or by parts per hour. Writing a non-zero value to CPSW\_GENF0\_PPM\_LOW\_REG\_I/ CPSW\_GENF0\_PPM\_HIGH\_REG\_I enables PPM operations. The PPM counter continually loads and decrements to zero and then loads again. A single CPTS\_RFT\_CLK adjustment is made when the PPM counter decrements to zero. The adjustment is up or down depending on the CPSW\_ESTF1\_CONTROL\_REG[0] PPM\_DIR bit. When PPM\_DIR bit is set a single CPTS\_RFT\_CLK time is subtracted from the generate function counter which has the effect of increasing the generate function frequency by the PPM amount. When PPM\_DIR bit is cleared a single CPTS\_RFT\_CLK time is added to the generate function counter which has the effect of decreasing the generate function frequency by the PPM amount.

#### Parts Per Million example:

To adjust for 100 parts per million the configured value for GENF\_PPM[41-0] (through CPSW\_GENF0\_PPM\_LOW\_REG\_I and CPSW\_GENF0\_PPM\_HIGH\_REG\_I) is:  
 $1,000,000/100 = 10,000(\text{decimal})$



### Parts Per Hour example:

To adjust for 1 part per hour at 1 GHz CPTS\_RFT\_CLK the configured value for GENF\_PPM[41-0] (through CPSW\_GENF0\_PPM\_LOW\_REG\_I and CPSW\_GENF0\_PPM\_HIGH\_REG\_I) is:

$(1,000,000,000\text{Hz}/1\text{pph}) * (3600 \text{ seconds/hour}) = 34630\text{B8A000}$  (hex)

### 9.4.27.12 Timestamp ESTFn

Each Ethernet port has a dedicated ESTFn generator which operates identically to the GENFn function.

### 9.4.27.13 Time Sync Events

Time Sync events are 96-bit values that are pushed onto the event FIFO and read by software in 32-bit reads. Four 32-bit registers, CPSW\_CPTS\_EVENT\_0\_REG through CPSW\_CPTS\_EVENT\_3\_REG hold the data of a time sync event. There are eight types of sync events:

- Time Stamp Push Event
- Time Stamp Counter Rollover Event (32-bit mode only)
- Time Stamp Counter Half-rollover Event (32-bit mode only)
- Hardware Time Stamp Push Event
- Ethernet Receive Event
- Ethernet Transmit Event
- Time Stamp Compare Event
- Host Transmit Event

Word or d	Bit fields																															
	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
event_0	time_stamp[31:0]																															
event_1	reserved	processor_number	event_type											message	sequence_id																	
event_2	reserved																								domain							
event_3	time_stamp[63:0]																															

Name	Description
time_stamp	Time Stamp – The timestamp is valid for transmit, receive, and time stamp push event types. The timestamp value is not valid for counter roll event types.

Name	Description
<b>port_number</b>	Port Number – indicates the port number of an Ethernet event (1 to 4 encoded) or the hardware push number (1 to 8 encoded).
<b>event_type</b>	Time Sync Event Type 0000 – Time Stamp Push Event 0001 – Time Stamp Rollover Event (32-bit mode only) 0010 – Time Stamp Half Rollover Event (32-bit mode only) 0011 – Hardware Time Stamp Push Event 0100 – Ethernet Receive Event 0101 – Ethernet Transmit Event 0110 – Time Stamp Compare Event 0111 – Host Event 1000 --- - reserved 1111
<b>message_type</b>	Message type – The message type value that was contained in an Ethernet transmit or receive time sync packet. This field is valid only for Ethernet transmit or receive events.
<b>sequence_id</b>	Sequence ID – The 16-bit sequence id is the value that was contained in an Ethernet transmit or receive time sync packet. This field is valid only for Ethernet transmit or receive events.
<b>domain</b>	Domain – The 8-bit domain is the value that was contained in an Ethernet transmit or receive time sync packet. This field is valid only for Ethernet transmit or receive events.
<b>preempt_queue</b>	Preempt Queue – 0 – The packet was received/transmitted on the express queue. 1 – The packet was received/transmitted on the preempt queue.

#### 9.4.27.13.1 Time Stamp Push Event

Software can obtain the current time stamp value (at the time of the write) by initiating a time stamp push event. The push event is initiated by setting the TS\_PUSH bit of the CPSW\_CPTS\_TS\_PUSH\_REG register. The time stamp value is returned in the event, along with a time stamp push event code. The upper 32-bits (CPSW\_CPTS\_EVENT\_3\_REG register) of the timestamp are zero in 32-bit mode.

#### 9.4.27.13.2 Time Stamp Counter Rollover Event (32-bit mode only)

The CPTS module contains a 32-bit time stamp value (CPSW\_CPTS\_EVENT\_0\_REG). The counter upper bits are maintained by host software. The rollover event indicates to software that the time stamp counter has rolled over from 0xFFFF FFFF to 0x0000 0000 and the software-maintained upper count value should be incremented. This event occurs only in 32-bit mode.

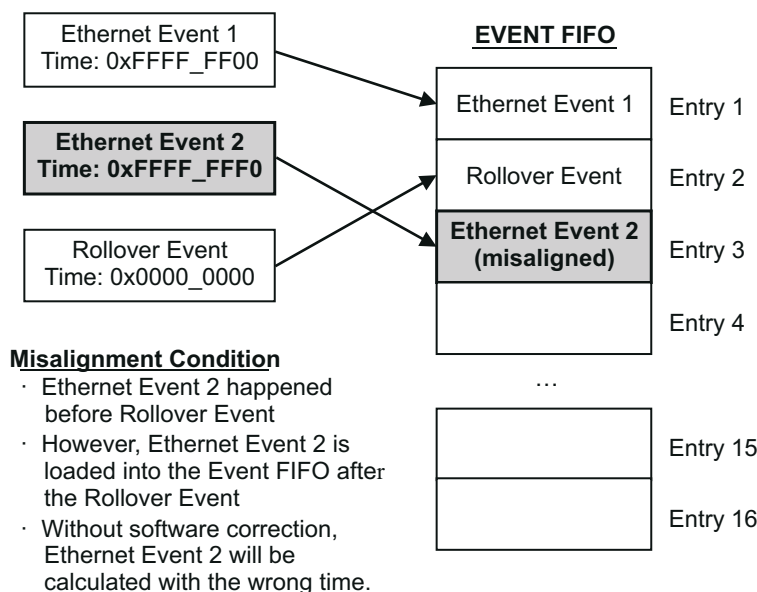
#### 9.4.27.13.3 Time Stamp Counter Half-rollover Event (32-bit mode only)

The CPTS includes a time stamp counter half-rollover event. The half-rollover event indicates to software that the time stamp value (CPSW\_CPTS\_EVENT\_0\_REG[31:0] TIME\_STAMP) has incremented from 0x7FFF FFFF to 0x8000 0000. The half-rollover event is included to enable software to correct a misaligned event condition. This event occurs only in 32-bit mode.

The half-rollover event is included to enable software to determine the correct time for each event that contains a valid time stamp value, such as an Ethernet event. If an Ethernet event occurs around a counter rollover (full rollover), the rollover event could possibly be loaded into the event FIFO before the Ethernet event, even though the Ethernet event time was actually taken before the rollover. [Figure 9-11](#) shows a misalignment condition. This misaligned event condition arises because an Ethernet event time stamp occurs at the beginning of a packet and time passes before the packet is determined to be a valid synchronization packet. The misaligned event condition occurs if the rollover occurs in the middle, after the packet time stamp has been taken, but before the packet has been determined to be a valid time sync packet.

Host software must detect and correct for misaligned event conditions. For every event time stamp after a rollover and before a half-rollover, software must examine the time stamp most significant bit. If bit 31 of the time stamp value is low (0x0000 0000 through 0x7FFF FFFF), then the event time stamp was taken after the rollover and no correction is required. If the value is high (0x8000 0000 through 0xFFFF FFFF), the time stamp value was taken before the rollover and a misalignment is detected. The misaligned case indicates to software that it must subtract one from the upper count value stored in software to calculate the correct time for the misaligned event. The misaligned event occurs only on the rollover boundary and not on the half-rollover boundary. Software only needs to check for misalignment from a rollover event to a half-rollover event.

When a rollover occurs, software increments the software time stamp upper value. The misaligned case indicates to software that the misaligned event time stamp has a valid upper value that is pre-increment, so one must be subtracted from the upper value to allow software to calculate the correct time for the misaligned event.



cpsw-014

**Figure 9-11. Event FIFO Misalignment Condition**

#### 9.4.27.13.4 Hardware Time Stamp Push Event

There are four hardware time stamp inputs (CPTS\_HW[1:4]\_TS\_PUSH events) that can cause hardware time stamp push events to be loaded into the Event FIFO. Each time stamp input is mapped in the device as shown in Figure 9-7. The event is loaded into the event FIFO on the rising edge of the timer, and the PORT\_NUMBER field in the CPSW\_CPTS\_EVENT\_1\_REG register indicates the hardware push input that caused the event (encoded).

The hardware time stamp inputs are asynchronous and are low frequency signals. The CPTS logic synchronizes and performs a rising edge detect on the incoming asynchronous input.

Each hardware time stamp input must be asserted for at least 10 periods of the selected CPTS\_RFT\_CLK clock. Each input can be enabled or disabled by setting the respective bits in the CPSW\_CPTS\_CONTROL\_REG register.

Hardware time stamps are intended to be an extremely low frequency signals, such that the event FIFO does not overrun. Software must keep up with the event FIFO and ensure that there is no overrun, or events will be lost.

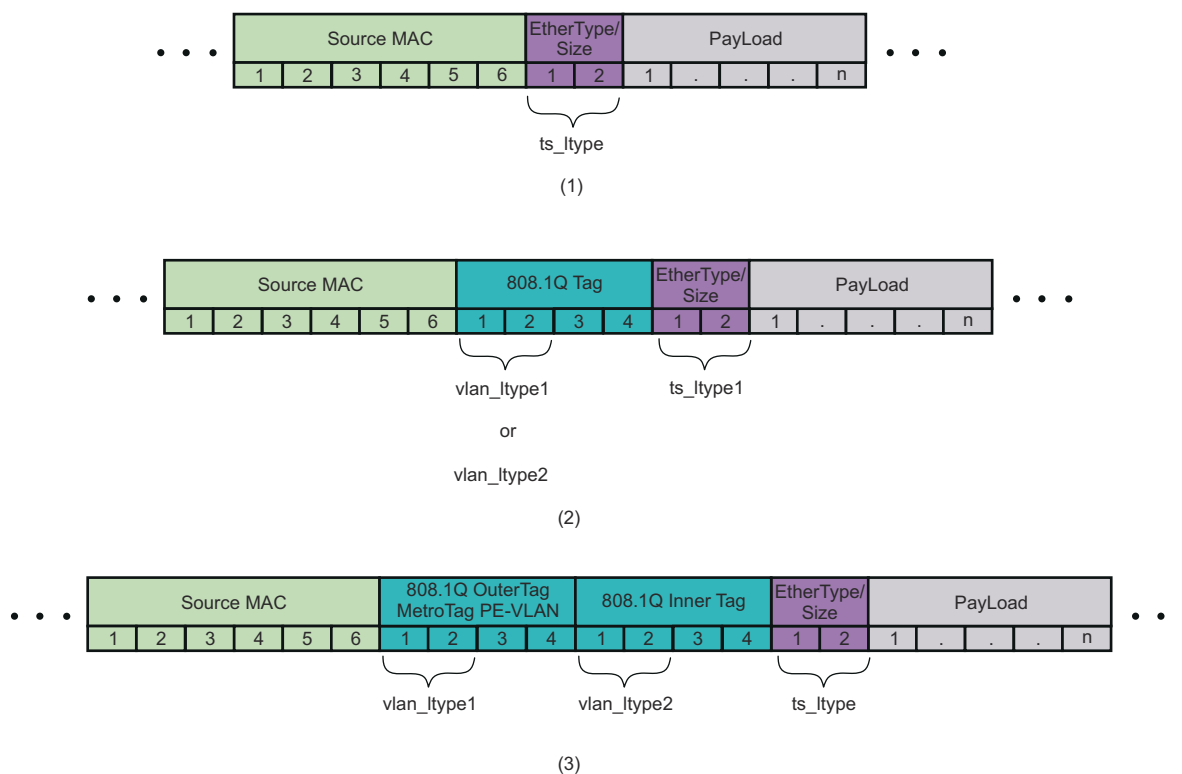
### 9.4.27.13.5 Ethernet Port Events

Packets transmitted or received on each Ethernet port can generate Ethernet Transmit Events or Ethernet Receive Events, respectively. The CPTS hardware will decode each packet to determine if it is a valid CPTS time sync event.

According to the IEEE 802.3 Ethernet standard, each Ethernet frame contains a 2-octet EtherType field to indicate which protocol is encapsulated in the PayLoad field, as shown in Figure 9-12. For standard time sync packets, this will contain the EtherType for the Precision Time Protocol (IEEE 1588), which is defined as 0x88F7. The CPTS hardware will compare this field to the TS\_LTYPE1 field in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register or the TS\_LTYPE2 field in CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register (depending on which enable bit was set) , which should also be programmed to 88F7h.

When a virtual LAN is used, an additional 4-octet 802.1Q tag is inserted in the Ethernet frame before the EtherType field, as shown in Figure 9-12. To indicate to the CPTS hardware that a virtual LAN is in use, the TS\_TX\_VLAN\_LTYPE1\_EN (or TS\_TX\_VLAN\_LTYPE2\_EN) enable bit must be set in the CPSW\_PN\_TS\_CTL\_REG register. The EtherType for the 802.1Q tag is defined as 0x8100, and the CPTS hardware will compare this value to the TS\_VLAN\_LTYPE1 (or TS\_VLAN\_LTYPE2 depending on which enable bit was set) field in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register, which should also be programmed to 0x8100.

When two stacked VLANs are used, two additional 4-octet 801.Q tags are inserted in the Ethernet frame before the EtherType field, as shown in Figure 9-12. In this case, both TS\_VLAN\_LTYPE1 and TS\_VLAN\_LTYPE2 must be enabled. The outer tag must match the value of the TS\_VLAN\_LTYPE1 field, and the inner tag must match the value of the TS\_VLAN\_LTYPE2 field.



cpsw-015

**Figure 9-12. Partial Ethernet-II Frames Showing Register Mapping of EtherTypes for a Simple Frame (1), a Single 1Q Tag Added (2), and Two 1Q Tags Added (3)**

#### 9.4.27.13.5.1 Ethernet Port Receive Event

This section describes Ethernet port receive events. Ethernet port generates time synchronization events for valid received time sync packets. For every packet received on the Ethernet port, a timestamp will be captured by the receive module inside the CPTS for the corresponding port. The time stamp will be captured by the receive module regardless of whether or not the packet is a time synchronization packet to make sure that the time stamp is captured as soon as possible. The packet is sampled on both the rising and falling edges of the CPTS\_RFT\_CLK, and the time stamp will be captured once the start of frame delimiter for the receive packet is detected.

After the time stamp has been captured, the receive interface will begin parsing the packet to determine if it is a valid Ethernet time synchronization packet. The CPSW decoder determines if the packet is a valid Ethernet receive time synchronization event. The receive interface for the port will use the following criteria to determine if the packet is a valid Annex D, Annex E, or Annex F time synchronization Ethernet receive event:

##### **Annex D (IPv4)**

1. Receive annex D time sync is enabled (TS\_RX\_ANNEX\_D\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register).
2. One of the sequences below is true.
  - a. The first packet LTYPE matches 0x0800
  - b. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches 0x0800
  - c. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches 0x0800
  - d. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches 0x0800
3. Byte 14 (the byte after the LTYPE) contains 0x45 (IPv4).

---

##### **Note**

The byte numbering assumes that there are no VLANs. The byte number is intended to show the relative order of the bytes.

- 
4. Byte 20 contains 0bXXX00000 (5 lower bits zero) and Byte 21 contains 0x00 (fragment offset zero)
  5. Byte 22 contains 0x01 (HOP Limit = 1) if the TS\_TTL\_NONZERO bit in the switch CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is cleared to 0h, or byte 22 contains any value if CPSW\_PN\_TS\_CTL\_LTYPE2\_REG is set to 1h. Byte 22 is the TTL/HOP field.
  6. Byte 23 contains 0x11 (Next Header UDP Fixed).
  7. The TS\_UNI\_EN bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is cleared to 0h and Bytes 30 through 33 contain:
    - a. Decimal 224.0.1.129 and the TS\_129 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - b. Decimal 224.0.1.130 and the TS\_130 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - c. Decimal 224.0.1.131 and the TS\_131 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - d. Decimal 224.0.1.132 and the TS\_132 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - e. Decimal 224.0.0.107 and the TS\_107 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set

-OR-

The TS\_UNI\_EN bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set and Bytes 30 through 33 contain any values.

8. Bytes 36 and 37 contain:





9. The packet message type is enabled in the MSG\_TYPE\_EN field in the CPSW\_PN\_TS\_CTL2\_REG register.
10. The packet was received without error (not long/short/mac\_ctl/CRC/code/align).

### **Annex F (IEEE 802.3)**

1. Receive Annex F time sync is enabled (TS\_RX\_ANNEX\_F\_EN is set in the switch CPSW\_PN\_TS\_CTL\_REG register).
2. One of the sequences below is true:
  - a. The first packet LTYPE matches TS\_LTYPE1 in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register. LTYPE 1 should be used when only one time sync LTYPE is to be enabled.
  - b. The first packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register.
  - c. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE1 in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register
  - d. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and TS\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register.
  - e. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE1 in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register.
  - f. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and TS\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register.
  - g. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches TS\_LTYPE1 in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register.
  - h. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_RX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and TS\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register
3. The PTP message begins in the byte after the LTYPE.
4. The packet message type is enabled in the TS\_MSG\_TYPE\_EN field in the CPSW\_PN\_TS\_CTL\_REG register.
5. The packet was received without error (not long/short/mac\_ctl/CRC/code/align).

If all of the criteria described above are met for either Annex D, Annex E, or Annex F, and the packet is determined to be a valid time synchronization packet, then the RX interface will push an Ethernet receive event into the event FIFO.

#### **9.4.27.13.5.2 Ethernet Port Transmit Event**

This section describes Ethernet port transmit events. For every packet transmitted on the Ethernet ports, the port transmit interface will begin parsing the packet to determine if it is a valid Ethernet time synchronization packet. The CPTS transmit interface for the port will use to the following criteria to determine if the packet is a valid time synchronization Ethernet transmit event. The CPSW decoder determines if the packet is a valid ethernet receive time synchronization event. To be a valid Ethernet transmit time synchronization event, the conditions listed below must be true for either Annex D, Annex E, or Annex F:

## Annex D (IPv4)

1. Transmit time sync is enabled (TS\_TX\_ANNEX\_D\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register).
2. One of the sequences below is true.
  - a. The first packet LTYPE matches 0x0800
  - b. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches 0x0800
  - c. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches 0x0800
  - d. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches 0x0800
3. Byte 14 (the byte after the LTYPE) contains 0x45 (IPv4).

---

### Note

The byte numbering assumes that there are no VLANs. The byte number is intended to show the relative order of the bytes.

- 
4. Byte 20 contains 0bXXX00000 (5 lower bits zero) and Byte 21 contains 0x00 (fragment offset zero)
  5. Byte 22 contains 0x01 (HOP Limit = 1) if the TS\_TTL\_NONZERO bit in the switch CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is cleared to 0h, or byte 22 contains any value if TS\_TTL\_NONZERO is set to 1h. Byte 22 is the TTL/HOP field.
  6. Byte 23 contains 0x11 (Next Header UDP Fixed).
  7. The TS\_UNI\_EN bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is cleared to 0h and Bytes 30 through 33 contain:
    - a. Decimal 224.0.1.129 and the TS\_129 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - b. Decimal 224.0.1.130 and the TS\_130 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - c. Decimal 224.0.1.131 and the TS\_131 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - d. Decimal 224.0.1.132 and the TS\_132 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - e. Decimal 224.0.0.107 and the TS\_107 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - f. The TS\_UNI\_EN bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set and Bytes 30 through 33 contain any values.
  8. Bytes 36 and 37 contain:
    - a. Decimal 0x01 and 0x3F respectively and the TS\_319 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - b. Decimal 0x01 and 0x40 respectively and the TS\_320 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set.
  9. The PTP message begins in byte 42.
  10. The packet message type is enabled in the TS\_MSG\_TYPE\_EN field in the CPSW\_PN\_TS\_CTL\_REG register.
  11. The packet was sent by host port 0.

## Annex E (IPv6)

1. Transmit annex E time sync is enabled (TS\_TX\_ANNEX\_E\_EN bit is set in the switch CPSW\_PN\_TS\_CTL\_REG register).
2. One of the sequences below is true.
  - a. The first packet LTYPE matches 0x86dd.



- b. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches 0x86dd
  - c. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches 0x86dd
  - d. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches 0x86dd
3. Byte 14 (the byte after the LTYPE) contains 0x6X (IPv6).
  4. Byte 20 contains 0x11 (UDP Fixed Next Header).
  5. Byte 21 contains 0x01 (Hop Limit = 1) if the TS\_TTL\_NONZERO bit in the switch CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is cleared to 0h, or byte 21 contains any value if TS\_TTL\_NONZERO is set to 1h. Byte 21 is the TTL/HOP field..
  6. The TS\_UNI\_EN bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is cleared to 0 and Bytes 38 through 53 contain:
    - a. FF0M:0:0:0:0:0:0:0181 and the TS\_129 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - b. FF0M:0:0:0:0:0:0:0182 and the TS\_130 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - c. FF0M:0:0:0:0:0:0:0183 and the TS\_131 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - d. FF0M:0:0:0:0:0:0:0184 and the TS\_132 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
    - e. FF0M:0:0:0:0:0:0:006B and the TS\_107 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set

#### Note

All values above are 16-bit hex numbers where M is enabled in the TS\_MCAST\_TYPE\_EN field in the CPSW\_PN\_TS\_CTL2\_REG register.

-OR-

The TS\_UNI\_EN bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set to 1h and Bytes 38 through 53 contain any value.

7. Bytes 56 and 57 contain (UDP Header in bytes 54 through 61):
  - a. Decimal 0x01 and 0x3F respectively and the TS\_319 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set, or
  - b. Decimal 0x01 and 0x40 respectively and the TS\_320 bit in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register is set.
8. The PTP message begins in byte 62.
9. The packet message type is enabled in the TS\_MSG\_TYPE\_EN field in the CPSW\_PN\_TS\_CTL\_REG register.
10. The packet was sent by host port 0.

#### Annex F (IEEE 802.3)

1. Transmit time sync is enabled (TS\_TX\_ANNEX\_F\_EN is set in the switch CPSW\_PN\_TS\_CTL\_REG register).
2. One of the sequences below is true:
  - a. The first packet LTYPE matches TS\_LTYPE1 in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register. LTYPE 1 should be used when only one time sync LTYPE is to be enabled.
  - b. The first packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and TS\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register.

- c. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register.
  - d. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE1 in the CPSW\_PN\_TS\_SEQ\_LTYPE\_REG register.
  - e. The first packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register.
  - f. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and TS\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register.
  - g. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register.
  - h. The first packet LTYPE matches TS\_VLAN\_LTYPE1 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE1\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the second packet LTYPE matches TS\_VLAN\_LTYPE2 in the CPSW\_PN\_TS\_VLAN\_LTYPE\_REG register and TS\_TX\_VLAN\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register and the third packet LTYPE matches TS\_LTYPE2 in the CPSW\_PN\_TS\_CTL\_LTYPE2\_REG register and TS\_LTYPE2\_EN is set in the CPSW\_PN\_TS\_CTL\_REG register.
3. The packet message type is enabled in the TS\_MSG\_TYPE\_EN field in the CPSW\_PN\_TS\_CTL\_REG register.
  4. The packet was sent by host port 0.

If all of the criteria described above are met, and the packet is determined to be a valid time synchronization packet, then the time stamp for the transmit event will not be generated until the start of frame delimiter of the packet is actually transmitted. The start of frame delimiter will be sampled on every rising and falling edge of the CPTS\_RFT\_CLK. Once the packet is transmitted, then the TX interface will push an Ethernet transmit event into the event FIFO.

#### 9.4.27.13.5.3

**Table 9-23. Values of Message Type Field**

Message Type	Value (hex)
Sync	0
Delay_Req	1
Pdelay_Req	2
Pdelay_Resp	3
Reserved	4:7
Follow_Up	8
Delay_Resp	9
Pdelay_Resp_Follow_Up	A
Announce	B
Signaling	C
Management	D
Reserved	E:F

Once a transmitted or received packet is determined to be a valid time sync packet, the Ethernet Transmit Event or Ethernet Receive Event is loaded onto the Event FIFO.

The CPTS\_EVENT\_1 register contains the Message Type and Sequence ID values from the original time sync packet. The CPTS\_EVENT\_0 (and CPTS\_EVENT\_3) register contains the time stamp value when the packet arrived at the corresponding port.

#### 9.4.27.14 Timestamp Compare Event

##### Note

Timestamp compare events are generated for non-toggle mode only.

The CPTS can generate an event for a time stamp comparison in 32-bit or 64-bit mode.

##### 9.4.27.14.1 32-Bit Mode

The CPTS\_COMP output is also asserted when the event is generated. The event is generated when the 32-bit time stamp value (EVENT\_0\_REG) compares with the CPSW\_CPTS\_TS\_COMP\_VAL\_REG register and the CPSW\_CPTS\_TS\_COMP\_LEN\_REG value is non-zero. The CPSW\_CPTS\_TS\_COMP\_LEN\_REG value should be written by software after the CPSW\_CPTS\_TS\_COMP\_VAL\_REG register is written and should be zero when the comparison value is written.

##### 9.4.27.14.2 64-Bit Mode

The CPTS\_COMP output is also asserted when the event is generated. The event is generated when the 64-bit time stamp value (CPSW\_CPTS\_EVENT\_0\_REG and CPSW\_CPTS\_EVENT\_3\_REG) compares with the CPWS0\_TS\_COMP\_VAL\_REG and CPWS0\_TS\_COMP\_HIGH\_VAL\_REG registers and the CPWS0\_TS\_COMP\_LEN\_REG value is non-zero. The CPWS0\_TS\_COMP\_LEN\_REG value should be written by software after the CPWS0\_TS\_COMP\_VAL\_REG register is written and should be zero when the comparison value is written.

#### 9.4.27.15 Host Transmit Event

The host can send a packet to be transmitted on an Ethernet port that will generate a time synchronization event. The host sets the TSTAMP\_EN bit and sends the DOMAIN, MESSAGE\_TYPE, and SEQUENCE\_ID in the additional control information that resides in the protocol specific section of the descriptor that is transmitted to the CPSW\_2G. An event is then generated and placed on the event FIFO once the packet is transmitted. Host events allow the user to timestamp exactly when a software generated packet exits the device.

#### 9.4.27.16 CPTS Interrupt Handling

The **TS\_PEND** interrupt is enabled by writing a logic high to the **ts\_int\_enable** bit in the interrupt enable register. The raw interrupt value (before the enable) can be read by reading the **TS\_Instat\_raw** register. The enabled interrupt value can be read by reading the **TS\_Instat\_masked** register.

Software can process time sync events via interrupts in the following way:

1. Enable the **TS\_PEND** interrupt by writing a logic high to bit zero of the **TS\_Int\_enable** register.
2. Upon interrupt, read the **Event\_0-4** register values.
3. Write logic high to bit-0 of the **Event\_Pop** register to pop the previously read value off of the event FIFO.
4. Process the end of interrupt as required by the upper level modules (outside the scope of CPTS).

Software has the option of processing more than a single event from the event FIFO in the interrupt service routine in the following way:

1. Enable the **TS\_PEND** interrupt by writing a logic high to bit zero of the **TS\_Int\_enable** register.
2. Upon interrupt enter the CPTS service routine.
3. Read the **Event\_0-4** register values.
4. Write a logic high to bit-0 of the **Event\_Pop** register to pop the previously read value off of the event FIFO.
5. Wait for an amount of time greater than four **RCLK** periods plus four **VBUSP\_CLK** periods.

6. Read bit 0 (**ts\_pend\_raw**) in the **TS\_Intstat\_raw** register to determine if another valid event is in the event FIFO. If bit zero is asserted then go to step 3. If bit 0 is not asserted then go to step 7.
7. Process the end of interrupt as required by the upper level modules (outside the scope of CPTS).

Software also has the option of disabling the interrupt and polling the **ts\_intstat\_raw** bit to determine if a valid event is on the event FIFO.

#### 9.4.28 MII Management Interface (MDIO)

The MII Management interface module implements the 802.3 serial management interface to interrogate and control external Ethernet PHY using a two-wire bus.

##### Features Supported

- Clause 22 and CLuse 45 support
- Supports up to 32 PHY addresses.
- Two user access registers to control and monitor up to two PHYs simultaneously.
- VBUS 3.0 compliant slave interface for configuration and control.
- Each PHY can be individually enabled to be polled.
- The inter-poll gap between PHY polls can be changed.
- State Change Mode of operation to monitor up to 32 PHYs simultaneously.
- The MDIO interface can be manually controlled by software for GPIO operations.

##### 9.4.28.1 MDIO Frame Formats

##### Clause 22

**Table 9-24. MDIO Clause 22 Read Frame Format**

Pre-amble	Start Delimiter	Operation Code	PHY Address	MMD Number	Turnaround	Data
FFFF FFFFh	01	10	AAAAA	RRRRR	Z0	DDDD.DDDD.DDDD.DDDD

**Table 9-25. MDIO Clause 22 Write Frame Format**

Pre-amble	Start Delimiter	Operation Code	PHY Address	MMD Number	Turnaround	Data
FFFF FFFFh	01	01	AAAAA	RRRRR	10	DDDD.DDDD.DDDD.DDDD

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor should pull the **MDIO** line to a logic one. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic one bits on the **MDIO** line with 32 corresponding cycles on **MDCLK** to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on **MDIO** with 32 corresponding **MDCLK** cycles before it responds to any other transaction.

##### Preamble

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a "1". This sequence provides the PHY a pattern to use to establish synchronization.

##### Start Delimiter

The preamble is followed by the start delimiter which is indicated by a "01" pattern. The pattern assures transitions from the default logic one state to zero and back to one.

##### Operation Code

The operation code for a read is "10", while the operation code for a write is a "01".

##### PHY Address

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSbit of the PHY address.

### Register Address

The Register address is 5 bits allowing 32 registers to be addressed within each PHY. Refer to the 10/100 PHY address map for addresses of individual registers.

### Turnaround

An idle bit time during which no device actively drives the MDIO signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

### Data

The Data field is 16-bits. The first bit transmitted and received is the MSbit of the data word.

## Clause 45

[Table 9-26](#) shows the address, [Table 9-27](#) shows the read format and [Table 9-28](#) shows the write format of the supported Clause 45 MII Management interface frames. Post-increment accesses are not supported.

**Table 9-26. MDIO Clause 45 Address Frame Format**

Pre-amble	Start Delimiter	Operation Code	PHY Address	MMD Number	Turnaround	Data
FFFF FFFFh	00	00	AAAAA	RRRRR	10	AAAA.AAAA.AAAA.AAAA

**Table 9-27. MDIO Clause 45 Read Frame Format**

Pre-amble	Start Delimiter	Operation Code	PHY Address	MMD Number	Turnaround	Data
FFFF FFFFh	00	11	AAAAA	RRRRR	Z0	DDDD.DDDD.DDDD.DDDD

**Table 9-28. MDIO Clause 45 Write Frame Format**

Pre-amble	Start Delimiter	Operation Code	PHY Address	MMD Number	Turnaround	Data
FFFF FFFFh	00	01	AAAAA	RRRRR	10	DDDD.DDDD.DDDD.DDDD

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor should pull the **MDIO** line to a logic one. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic one bits on the **MDIO** line with 32 corresponding cycles on **MDCLK** to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on **MDIO** with 32 corresponding **MDCLK** cycles before it responds to any other transaction. The **MDIO User\_Addr** registers must be written before a read or write operation is performed to set the address used in the operation. Each read or write operation has a preceeding address frame.

### Preamble

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a "1". This sequence provides the PHY a pattern to use to establish synchronization. The preamble is required in clause 45 operation.

### Start Delimiter

The preamble is followed by the start delimiter which is indicated by a "00" pattern.

### Operation Code

The operation code for an address transaction is "00", The operation code for a read is "11", while the operation code for a write is a "01".

## PHY Address

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSbit of the PHY address.

## MMD Number

The MMD number is the 5-bits allowing 32 unique values. The first bit transmitted is the MSbit.

## Turnaround

An idle bit time during which no device actively drives the MDIO signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

## Address

The address field is 16-bits on address operations. The first bit transmitted is the MSbit of the address word. Each read/write operation initiated has an automatic address operation initiated first that uses the **MDIO User\_Addr0/1** register values as the 16-bit address.

## Data

The Data field is 16-bits on read and write operations. The first bit transmitted and received is the MSbit of the data word.

### 9.4.28.2 MDIO Functional Description

The MDIO Management I/F will remain idle until enabled by setting the **enable** bit in the **MDIO Control** register. The module will then continuously poll the linkstatus bits from within the GenericStatusRegister of all enabled 32 PHY addresses. Individual PHY's can be enabled or disabled for polling thru the associated bit in the **Poll\_En** register. The **MDIO Link** and **MDIO alive** register bit values are updated on the poll of each PHY. In Normal Mode, The link status of two of the 32 possible PHY addresses can also be determined using the **MLINK** pin inputs. The **linksel** bit in the **MDIO USER\_PHY\_SEL** register determines the status input that is used. A change in the link status of the two PHYs being monitored will set the appropriate bit in the **MDIO LinkIntRaw** register and the **MDIO LinkIntMasked** register, if enabled by the **linkint\_enable** bit in the **MDIO User\_Phy\_Sel** register. In State Change Mode, a change in any PHY status will be indicated on the **MDIO LINKINT[0]** interrupt if enabled.

The **MDIO Alive** register is updated if the PHY acknowledged the read of the generic status register. In addition, any PHY register read transactions initiated by the host also update the **MDIO Alive** register bit associated with the PHY.

At any time, the host can initiate a transaction for the MDIO module to undertake using the **data**, **phyadr**, **regadd/MMD**, and **write** fields in an **MDIO User\_Access** register. When the host sets the **go** bit in this register, the MDIO module will begin the transaction without any further intervention from the host. Upon completion, the MDIO will clear the **go** bit and set the **userintraw** bit in the **MDIO User\_Int\_Raw** register corresponding to the **MDIO User\_Access** register being used. The corresponding bit in the **MDIO User\_Int\_Masked** register may also be set depending on the mask setting in the **MDIO User\_Int\_Mask\_Set** and **MDIO User\_Int\_Mask\_Clear** registers. A round-robin arbitration scheme is used to schedule transactions which may be queued by the host in different **MDIO User\_Access** registers. The host should check the status of the **go** bit in the **MDIO User\_Access** register before initiating a new transaction to ensure that the previous transaction has completed. The host can use the **ack** bit in the **MDIO User\_Access** register to determine the status of a read transaction.

Software may use the MDIO module to setup the auto-negotiation parameters of each PHY attached to a MAC port, retrieve the negotiation results, and setup the **MAC Control** register in the corresponding MAC.

### 9.4.29 Reset Isolation

The CPSW supports reset isolation of the Ethernet switch ports. When the **ISOLATE** input is asserted the below occur simultaneously:



- The Host Port 0 is removed from ALE processing (packets received on ports 1 through N-1 will be dropped to port 0).
- FHost packets are dropped. Any packet currently in progress when ISOLATE is asserted is dropped due to a FHost packet code error (and possible a CRC or FRAG error).
- THost packets in queue are dropped.

The intent of reset isolation is to allow packets to switch between the Ethernet ports while the remainder of the system is undergoing a reset. Isolation assumes that the external host and logic connected to the port 0 host interface is reset and will not be in the middle of a packet when **ISOLATE** is de-asserted.

#### 9.4.30 CPSW Initialization and Configuration

To configure the CPSW for operation the host must perform the following:

- Ensure that at least 2000 VBUSP\_CLK periods are run after reset is de-asserted.
- Configure the CPSW Control register.
- Configure the Ethernet Port Source Address registers
- Configure the Statistics Port Enable registers
- Configure the ALE.
- Configure the Ethernet Ports.

#### 9.4.31 Enet Mac Reset or XGMII/GMII Mode Change Configuration

- Set pn\_cmd\_idle in the Ethernet port Pn\_Mac\_Control registers.
- Wait for pn\_Idle to be indicated in the Ethernet port Pn\_Mac\_Status registers.
- Set pn\_soft\_reset in the Ethernet port Pn\_Soft\_Reset registers.
- Wait for pn\_soft\_reset in the Pn\_Soft\_Reset registers to be cleared to confirm reset completion.
- Configure the Ethernet ports.

#### 9.4.32 Memory Error Detection and Correction

The cpsw error detection and correction logic uses the IP ECC Aggregator Module. The ECC Aggregator allows the control of the cpsw RAMs as shown in the below table. ECC is always enabled. The cpsw FIFO RAMS implement ECC only on packet headers. The packet data is protected by Castagnoli CRC (regardless of the input packet CRC type or output CRC type). The ALE and EST RAMs have complete ECC as normal.

##### 9.4.32.1 Packet Header ECC

Only packet headers bits are protected by ECC in the cpsw RAMs. The **ECC\_Error\_Control1** register **ecc\_row** is not implemented – **ecc\_bit1** is implemented to determine which bit of the header is flipped for an SEC error when the **ecc\_crc\_mode** bit is cleared in the **CPSW\_Control** register. The ECC status registers return the RAM address that was flipped (**ecc\_row**) along with the **ecc\_bit1** value. Forcing double bit errors in testing can cause indeterminate switch operation if multiple used packet header bits are flipped given that only single bit errors are fixed by the ECC logic. Header bits 207 downto 200 are not currently used in the switch and may be used to test double bit errors without the possibility of requiring a reset for the switch to recover from the double bit error. No header bits are flipped when **ecc\_crc\_mode** is set to one. Either the **pn\_rx\_ecc\_err\_en** or the **pn\_tx\_ecc\_err\_en** bit must be set in the **PN\_Control** register to test ecc header errors.

The header ECC code is stored in bits 255 downto 208. If any bit is flipped in the ECC code, the flipped bit will be corrected, but the index of the flipped bit will be reported as bit zero. This implies that when the aggregator reports that there is a SEC on 'bit 0', it can mean two things: either SEC on data bit 0 or SEC somewhere inside the ECC code. Any packet header with ECC error issues a pulse on **ECC\_PULSE\_INTR** as does an ale ram ECC error.

##### 9.4.32.2 Packet Protect CRC

Each ingress packet without error is passed through the cpsw with an internally generated Castagnoli protect CRC. The protect CRC is checked on port egress for correctness and is replaced. If the CRC is correct (no RAM bit errors), then the packet is output with the selected port CRC type. If a protect CRC error is detected on THost then the **memory\_protect\_error** buffer descriptor bit will be asserted so that the packet is dropped to the host. If a protect CRC error is detected on Ethernet egress then the egress CRC will be generated

on the packet and at least one byte of the CRC will be inverted on output. CRC memory protect errors do not assert the **ECC\_PULSE\_INTR** signal. CRC memory protect errors are counted in the associated port statistics registers and issue an interrupt on **STAT\_PEND\_INTR** if any CRC memory protect error occurs (and the statistics for that port are enabled). When the **ecc\_crc\_mode** bit in the **CPSW\_Control** register is set, the **ecc\_bit1** aggregator register will flip the associated column bit in any FIFO memory read operation, inducing a CRC protect error when the protect CRC is checked. No header bits are flipped when **ecc\_crc\_mode** is set. Either the **pn\_rx\_ecc\_err\_en** or the **pn\_tx\_ecc\_err\_en** bit must be set in the **PN\_Control** register to test packet CRC errors.

RAM 0	ALE Ram
RAM 1	Port 0 FIFO Rx Ram
RAM 2	Port 0 FIFO Tx Ram
RAM 3	Port 1 FIFO Rx Ram
RAM 4	Port 1 FIFO Tx Ram
RAM 5	Port 2 FIFO Rx Ram
RAM 6	Port 2 FIFO Tx Ram
...	...
RAM (N*2) - 1	Port N FIFO Rx Ram
RAM (N*2)	Port N FIFO Tx Ram
RAM 19	EST Ram

### 9.4.32.3 Aggregator Ram Control

The ECC logic for each FIFO ram (receive and transmit) is divided into eight separate ECC encoders/decoders that encode/decode 26-bits of data each. Each of the 8 encoders (0 to 7) generates 6-bits of ECC code (48 code bits total), and each of the eight decoders (0 to 7) checks 6-bits of ECC code across the 26-bits of data (208 data bits total). The 48-bits of ECC code are passed through the ram in the upper 48 unused bits in the header word. The header data bits and ECC code bits are shown in the below table. The **ecc\_bit1[15:0]** value returned on error is a 16-bit value that is the concatenation of 5-bits of zero, 3-bits of the encoder/decoder number (0 to 7), 3-bits of zero, and 5-bits of index into the indicated 26-bit encoder/decoder. For example, an **ecc\_bit1** value of 0x0308 is bit-8 of encoder/decoder 3, which is header bit 86 ( $((26*3) + 8)$ ).

Header Data Bits	Encoder/Decoder
25:0	Encoder/Decoder 0 Data
51:26	Encoder/Decoder 1 Data
77:52	Encoder/Decoder 2 Data
103:78	Encoder/Decoder 3 Data
129:104	Encoder/Decoder 4 Data
155:130	Encoder/Decoder 5 Data
181:156	Encoder/Decoder 6 Data
207:182	Encoder/Decoder 7 Data
213:208	Encoder/Decoder 0 Code
219:214	Encoder/Decoder 1 Code
225:220	Encoder/Decoder 2 Code
231:226	Encoder/Decoder 3 Code
237:232	Encoder/Decoder 4 Code
243:238	Encoder/Decoder 5 Code
249:244	Encoder/Decoder 6 Code
255:250	Encoder/Decoder 7 Code



### 9.4.33 CPSW Network Statistics

The CPSW has statistics that record events associated with frame traffic on each port. STAT0 keeps statistics for PORT0, STAT1 for PORT1 and so on. All statistics are 32-bit registers. By convention the statistics registers are receive for ingress and transmit for egress for all ports.

If any bit in **stat\_port\_en** is set for a specific STAT module, the value written to a statistics register will be subtracted from the register value with the result being stored in the register. If a value greater than the statistics value is written, then zero will be written to the register (writing 0xffffffff will clear a statistics location). When all port enable bits are cleared to zero, all statistics registers are read/write (normal write direct, so writing 0x00000000 will clear a statistics location). All write accesses must be 32-bit accesses. In the below statistics descriptions, “the port” refers to any enabled port (with a corresponding set **stat\_port\_en** bit).

#### 9.4.33.1 Rx (only) Statistics Descriptions

##### 9.4.33.1.1 Good Rx Frames

The total number of good frames received on the port. A good frame is defined to be:

- any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

##### 9.4.33.1.2 Broadcast Rx Frames

The total number of good broadcast frames received on the port. A good broadcast frame is defined to be:

- any data or MAC control frame which was destined for address 0xffffffff only, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

##### 9.4.33.1.3 Multicast Rx Frames

The total number of good multicast frames received on the port. A good multicast frame is defined to be:

- any data or MAC control frame which was destined for any multicast address other than 0xffffffff, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

##### 9.4.33.1.4 Pause Rx Frames

The total number of IEEE 802.3X pause frames received by the port (whether acted upon or not). Such a frame:

- contained any unicast, broadcast, or multicast address, and
- contained the length/type field value 88.08 (hex) and the opcode 0x0001, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error, and
- pause frames are enabled on the port (pn\_tx\_flow\_en = 1).

The port could have been in either half or full-duplex mode.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic

or

The total number of priority based flow control (802.1Qbb) pause frames received by the port (whether acted upon or not). Such a frame:

- contained any unicast, broadcast, or multicast address, and
- contained the length/type field value 88.08 (hex) and the opcode 0x0001, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, and
- priority based flow control pause frames are enabled on the port.

See the **Rx CRC errors** statistic descriptions for definitions of CRC errors. Overruns have no effect upon this statistic

#### 9.4.33.1.5 Rx CRC Errors

The total number of frames received on the port that experienced a CRC error. Such a frame:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no code/align error, and
- had a CRC error.

Overruns have no effect upon this statistic.

A CRC error is defined to be:

- a frame containing an even number of nibbles, and
- fails the Frame Check Sequence test.

#### 9.4.33.1.6 Rx Align/Code Errors

The total number of frames received (ingress) on the port that experienced an alignment error or code error. Such a frame:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had either an alignment error or a code error.

Over-runs have no effect upon this statistic.

An alignment error is defined to be:

- a frame containing an odd number of nibbles, and
- also fails the Frame Check Sequence test if the final nibble is ignored.

A code error is defined to be:

- A frame which has been discarded because the port's **MRXER** pin driven with a one for at least one bit-time's duration at any point during the frame's reception.

---

#### Note

RFC 1757 etherStatsCRCAlignErrors Ref. 1.5 can be calculated by summing **Rx Align/Code Errors** and **Rx CRC errors** (see below).

---

10G: For XGMII, packets are ended at the code error and only that number of bytes are kept when rx\_cef is set (includes the code error bytes).

#### 9.4.33.1.7 Oversize Rx Frames

The total number of oversized frames received on the port. An oversized frame is defined to be:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was greater than pn\_rx\_maxlen in bytes, and

- had no CRC error, alignment error or code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.33.1.8 Rx Jabbers

The total number of jabber frames received on the port. A jabber frame is:

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was greater than pn\_rx\_maxlen bytes long, and
- had a CRC error, an alignment error, or a code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.33.1.9 Undersize (Short) Rx Frames

The total number of undersized frames received on the port. An undersized frame is defined to be:

- any data frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was less than 64 bytes long, and
- had no CRC error, alignment error or code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.33.1.10 Rx Fragments

The total number of frame fragments received on the port. A frame fragment is defined to be:

- any data frame (address matching does not matter), and
- was less than 64 bytes long, and
- had a CRC error, an alignment error, or a code error, and
- was not the result of a collision caused by half duplex, collision based flow control.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

#### 9.4.33.1.11 Rx IPG Error

The total number of 10G frames received on a port that had a correct preamble but did not have at least five bytes of IDLE preceding the frame. This does not indicate if the frame with the IPG error was kept or ignored.

#### 9.4.33.1.12 Rx Bottom Of FIFO Drop

### Ethernet Ports

The total number of ingress frames on a port that overran the port's receive FIFO and were dropped (bottom of receive FIFO). Port 0 (CPPI transmit port) should not drop packets on ingress because port 0 ingress flow control should be enabled. The Ethernet ports will only drop packets in the receive FIFO when receive flow control is enabled and the sending port ignores sent pause frame and then overruns the receive FIFO. The overrun frame is defined to be:

- any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- Was dropped on port 0 due to a lack of memory space in the receive FIFO

### Note

This stat should be zero

### Host Port 0

This statistic also counts frames dropped on port 0 that were 17 to 33 bytes (only for port 0). For Ethernet ports, the drop count for frames shorter than 33 bytes is included in the undersized or fragment count. Port 0 simply gives an indication that a packet with 33 bytes was dropped. No other statistics are counted for frames shorter than 33 bytes.

#### 9.4.33.1.13 Portmask Drop

##### All Ports

The total number of ingress frames on a port that were dropped by the ALE (the ALE did not forward the packet to any port). The frame was defined to be:

- any data or MAC control frame
- Was any length greater than 32 bytes
- Was dropped by the ALE - ale\_portmask = 0 (was not sent to any destination port)
- The frame could have been dropped due to error or other counted reason, so it could be counted elsewhere also.

---

##### Note

This stat does not count in the overall total as it includes every packet received greater than 32 bytes that had a zero **port\_mask**.

---

#### 9.4.33.1.14 Rx Top Of FIFO Drop

##### All Ports

The total number of frames received on a port that had a **START** of frame (SOF) overrun on any destination port egress (when attempting to load the packet from the top of the ingress port receive FIFO into any other port's transmit FIFO). If a multicast/broadcast packet is dropped by multiple destination ports then this statistic will increment by the number of ports that dropped the packet. Rx Top Of FIFO Drop is defined to be:

- any data or MAC control frame
- Was any length greater than 32 bytes
- Was dropped by the ALE - ale\_portmask = 0 (was not sent to any destination port)
- The frame could have been dropped due to error or other counted reason, so it could be counted elsewhere also.

#### 9.4.33.1.15 ALE Drop

##### All Ports

The total number of frames received on a port such that the destination address was not equal to the source address and the packet was not destined to the port it was received on, but the frame was not forwarded to any port (the **port\_mask** was zero).

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- the destination address was not equal to the source address, and
- The packet was not destined for the port it was receive on, and
- had a zero port\_mask

#### 9.4.33.1.16 ALE Overrun Drop

##### All Ports

The total number of frames received on a port that were dropped (zero **port\_mask**) due to exceeding the maximum ALE lookup rate (Port 0 should not have ALE Overrun Drops because the ingress rate is controlled to prevent it). This statistic should be zero and when non-zero indicates a system clock issue or indicates that short packets were sent with **pn\_rx\_csf\_en** at a rate that exceeded the maximum lookup rate.

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- the maximum ALE lookup rate was exceeded so the lookup was aborted and the packet was dropped.

#### 9.4.33.1.17 ALE Rate Limit Drop

##### All Ports

The total number of frames received on a port that were dropped (zero **port\_mask**) due to receive rate limiting on this port or due to transmit rate limiting on any destination port (not sent to all expected destination ports if transmit rate limiting).

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- the receive rate was exceeded and the packet was dropped, or the transmit rate was exceeded to any destination port and the packet was dropped to one or more expected destination ports (indicates that the destinations were pruned due to rate limiting).

#### 9.4.33.1.18 ALE VLAN Ingress Check Drop

The total number of frames received on a port that were dropped (zero **port\_mask**) due to VLAN ingress check failure.

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- the VLAN ID ingress check failed (the receive port was not in the group), and
- The address lookup did not return a match with the super bit set

#### 9.4.33.1.19 ALE DA=SA Drop

The total number of frames received on a port that were dropped (zero **port\_mask**) due to destination address equal to source address.

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- The destination address was equal to the source address
- The source address was not an entry in the table.

#### 9.4.33.1.20 Block Address Drop

The total number of frames received on a port that were dropped (zero **port\_mask**) due to the destination or source address being blocked.

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- the source or destination address matched a table entry with the block bit set.

#### 9.4.33.1.21 ALE Secure Drop

The total number of frames received on a port that were dropped (zero **port\_mask**) due to a secure violation (the source address is owned by a different receive port).

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and

- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- the source address is an entry in the table with the secure bit set and a port number for a different receive port.

#### **9.4.33.1.22 ALE Authentication Drop**

The total number of frames received on a port that were dropped (zero **port\_mask**) due to authentication failure.

- was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was any length (including <64 bytes and > pn\_rx\_maxlen bytes), and
- had no CRC error, alignment error or code error, and
- enable\_auth\_mode is set, and
- the source address is not equal to the destination address, and
- the source address is not a table entry, and
- the destination address is not a table entry with the super bit set.

#### **9.4.33.1.23 ALE Unknown Unicast**

The total number of frames received on a port that had a unicast destination address with an unknown source address. The frame is defined to be:

- was any data frame with a unicast destination address, and
- the source address was not a table entry, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

---

#### **Note**

The ALE Unknown Unicast Bytecount statistic is the number of bytes contained in the ALE Unknown Unicast frames.

---

#### **9.4.33.1.24 ALE Unknown Multicast**

The total number of frames received on a port that had a multicast destination address with an unknown source address. The frame is defined to be:

- was any data frame with a multicast destination address, and
- the source address was not a table entry, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

---

#### **Note**

The ALE Unknown Multicast Bytecount statistic is the number of bytes contained in the ALE Unknown Multicast frames.

---

#### **9.4.33.1.25 ALE Unknown Broadcast**

The total number of frames received on a port that had a Broadcast destination address with an unknown source address. The frame is defined to be:

- was any data frame with a broadcast destination address, and
- the source address was not a table entry, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

---

#### **Note**

The ALE Unknown Broadcast Bytecount statistic is the number of bytes contained in the ALE Unknown Broadcast frames.

---

#### 9.4.33.1.26 ALE Policer Match

The total number of frames received on a port that had a matched a policer. The frame is defined to be:

- any data frame, and
- matched a condition on a policer, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

#### 9.4.33.1.27 ALE Policer Match Red

The total number of frames received on a port that had matched a policer and the condition was red. The frame is defined to be:

- any data frame, and
- matched a policer with the condition red, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

#### 9.4.33.1.28 ALE Policer Match Yellow

The total number of frames received on a port that had matched a policer and the condition was yellow. The frame is defined to be:

- any data frame, and
- matched a policer with the condition yellow, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

#### 9.4.33.1.29 IET Receive Assembly OK

The total number of correctly received and re-assembled preemptable frames.

- any preemptable frame received
- was any size, and
- was correctly received and re-assembled without error.

#### 9.4.33.1.30 IET Receive Assembly Error

The total number of preemptable received frames with IET assembly errors.

- any frame received
- was any size, and
- was a non-initial fragment that mismatched the frame count or fragment count (went to the assembly error state in the IET receive state machine).

#### 9.4.33.1.31 IET Receive SMD Error

The total number of received frames rejected due to an unknown SMD value or received frames rejected with an SMD-C when no frame is in progress.

- any frame received
- was any size, and
- was rejected because of an unknown SMD value or SMD-C with no frame in progress.

---

#### Note

If **iet\_en** is not set, this statistic counts any received frame with any non express SMD.

---

#### 9.4.33.1.32 IET Receive Merge Fragment Count

The total number of received non-initial fragments that did not have an assembly error. The IET stat aMACMergeFragCountRx is derived by adding the Receive Assembly Error count to this value.

- any frame received
- was any size, and
- was a non-initial fragment that did not contain an assembly error.



### 9.4.33.1.33 Rx Octets

The total number of bytes in all good frames received on the port. A good frame is defined to be:

- any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode, and
- was of length 64 to pn\_rx\_maxlen bytes inclusive, and
- had no CRC error, alignment error or code error.

See the **Rx Align/Code Errors** and **Rx CRC errors** statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.

### 9.4.33.2 Tx (only) Statistics Descriptions

The maximum and minimum transmit frame size is software controllable.

Thost overruns (**P0\_Tx\_SOF\_OVERRUN** and **P0\_Tx\_MOF\_OVERRUN**) have no effect on Tx statistics. They are counted separately.

#### 9.4.33.2.1 Good Tx Frames

The total number of good frames transmitted on the port. A good frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- was any length, and
- had no late or excessive collisions, no carrier loss and no underrun.

#### 9.4.33.2.2 Broadcast Tx Frames

The total number of good broadcast frames transmitted on the port. A good broadcast frame is defined to be:

- any data or MAC control frame destined for address 0xFFFFFFFF only, and
- was of any length, and
- had no late or excessive collisions, no carrier loss and no underrun

#### 9.4.33.2.3 Multicast Tx Frames

The total number of good multicast frames transmitted on the port. A good multicast frame is defined to be:

- any data or MAC control frame destined for any multicast address other than 0xFFFFFFFF, and
- was of any length, and
- had no late or excessive collisions, no carrier loss and no underrun.

#### 9.4.33.2.4 Pause Tx Frames

This statistic indicates the number of IEEE 802.3X pause frames transmitted by the port.

Pause frames cannot contain a CRC error because they are created in the transmitting MAC, so these error conditions have no effect upon the statistic. Pause frames sent by software will not be included in this count.

Since pause frames are only transmitted in full duplex carrier loss and collisions have no effect upon this statistic.

Transmitted pause frames are always 64 byte multicast frames so will appear in the **Tx Multicast Frames** and **64octect Frames** statistics.

#### 9.4.33.2.5 Collisions

This statistic records the total number of times that the port experienced a collision. Collisions occur under two circumstances.

1. When a transmit data or MAC control frame:
  - was destined for any unicast, broadcast or multicast address, and
  - was any size, and
  - had no carrier loss and no underrun, and
  - experienced a collision. A jam sequence is sent for every non-late collision, so this statistic will increment on each occasion if a frame experiences multiple collisions (and increments on late collisions).



CRC errors have no effect upon this statistic.

2. When the port is in half-duplex mode, flow control is active, and a frame reception begins.

#### **9.4.33.2.6 Single Collision Tx Frames**

The total number of frames transmitted on the port that experienced exactly one collision. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced one collision before successful transmission. The collision was not late.

CRC errors have no effect upon this statistic.

#### **9.4.33.2.7 Multiple Collision Tx Frames**

The total number of frames transmitted on the port that experienced multiple collisions. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced 2 to 15 collisions before being successfully transmitted. None of the collisions were late.

CRC errors have no effect upon this statistic.

#### **9.4.33.2.8 Excessive Collisions**

The total number of frames for which transmission was abandoned due to excessive collisions. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced 16 collisions before abandoning all attempts at transmitting the frame. None of the collisions were late.

CRC errors have no effect upon this statistic.

#### **9.4.33.2.9 Late Collisions**

The total number of frames on the port for which transmission was abandoned because they experienced a late collision. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- experienced a collision later than 512 bit-times into the transmission. There may have been up to 15 previous (non-late) collisions which had previously required the transmission to be re-attempted. The Late Collisions statistic dominates over the single, multiple and excessive Collisions statistics - if a late collision occurs the frame will not be counted in any of these other three statistics.

CRC errors, carrier loss, and underrun have no effect upon this statistic.

#### **9.4.33.2.10 Deferred Tx Frames**

The total number of frames transmitted on the port that first experienced deferment. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no carrier loss and no underrun, and
- experienced no collisions before being successfully transmitted, and
- found the medium busy when transmission was first attempted, so had to wait.

CRC errors have no effect upon this statistic

See RFC1623 Ref. 2.6 dot3StatsDeferredTransmissions.

#### **9.4.33.2.11 Carrier Sense Errors**

The total number of frames on the port that experienced carrier loss. Such a frame:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address, and
- was any size, and
- the carrier sense condition was lost or never asserted when transmitting the frame (the frame is not retransmitted). This is a transmit only statistic. Carrier Sense is a don't care for received frames. Transmit frames with carrier sense errors are sent until completion and are not aborted.
- CRC errors and underrun have no effect upon this statistic.

#### 9.4.33.2.12 Tx Octets

The total number of bytes in all good frames transmitted on the port. A good frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- was any size, and
- had no late or excessive collisions and no carrier loss.

#### 9.4.33.2.13 Transmit Priority 0-7

The total number of frames transmitted on the port from transmit FIFO priority 0-7. Collision retries do not affect this statistic. Pause frames do not affect this statistic.

- any frame transmitted from priority 0-7, and
- was less than or equal to `cpsw_tx_pri(0-7)_maxlen`.
- Collision retries are not counted in this statistic
- Pause frames are not counted in this statistic.
- Carrier sense errors do not affect this statistic.

---

#### Note

The Transmit Priority 0-7 Bytecount statistic is the number of bytes contained in the frames of the Transmit Priority 0-7 statistic.

---

#### 9.4.33.2.14 Transmit Priority 0-7 Drop

The total number of transmit frames on the port that overran the transmit FIFO priority 0-7 and were dropped. This count includes frames dropped due to `cpsw_tx_pri(0-7)_maxlen`.

- any frame destined to be transmitted from priority 0-7, and
- was any size, and
- was dropped due to priority 0-7 FIFO overrun (Start of packet overrun).
- Was dropped due to frame size larger than `cpsw_tx_pri(0-7)_maxlen`.

---

#### Note

The Transmit Priority 0-7 Drop Bytecount statistic is the number of bytes contained in the frames of the Transmit Priority 0-7 Drop statistic.

---

#### 9.4.33.2.15 Transmit Memory Protect Errors

The total number of transmit frames on the port that had a memory protect CRC error on egress.

- any frame destined to be transmitted, and
- was any size, and
- Had a memory protect CRC error on egress.

Frames to the host with memory protect errors are indicated to be dropped with a set receive buffer descriptor **drop** bit. Ethernet frames will have at least one byte of the generated port type CRC inverted on egress.

This statistic is 8-bits wide only and will not rollover but will limit at 0xff.

A non-zero value in this statistic will issue a **STAT\_PEND\_INTR** interrupt for the associated port.

#### 9.4.33.2.16 IET Transmit Merge Fragment Count

The total number of non-initial preemptable transmit fragments on preemptable transmit.

- any frame destined to be transmitted on the preemptable port, and
- was any size, and
- was a non-initial fragment.

#### 9.4.33.2.17 IET Transmit Merge Hold Count

The total number of preemptable frames that were preempted and reassembled by the assertion of **pn\_mac\_hold** in the **Enet\_Pn\_IET\_Control** register or were preempted by EST. The IET statistic aMACMergeHoldCount can be derived and maintained by software.

- any frame destined to be transmitted on the preemptable port, and
- was any size, and
- was preempted by the assertion of pn\_mac\_hold, or
- was preempted by Enhanced Scheduled Traffic (EST).

### 9.4.33.3 Rx and Tx (shared) Statistics Descriptions

#### 9.4.33.3.1 Net Octets

The total number of bytes of frame data received and transmitted on the port. Each frame counted:

- was any data or MAC control frame destined for any unicast, broadcast or multicast address (address match does not matter), and
- was of any size (including <64 byte and > pn\_rx\_maxlen byte frames).

Also counted in this statistic are:

- every byte transmitted before a carrier-loss was experienced,
- every byte transmitted before each collision was experienced, (i.e. multiple retries are counted each time),
- every byte received if the port is in half-duplex mode until a jam sequence was transmitted to initiate flow control. (The jam sequence was not counted to prevent double-counting).

Error conditions such as alignment errors, CRC errors, code errors, overruns and underruns do not affect the recording of bytes by this statistic.

The objective of this statistic is to give a reasonable indication of Ethernet utilization.

#### 9.4.33.3.2 Rx + Tx 64 Octet Frames

The total number of 64-byte frames received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was exactly 64 bytes long. (If the frame was being transmitted and experienced carrier loss that resulted in a frame of this size being transmitted, then the frame will be recorded in this statistic).

CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.

#### 9.4.33.3.3 Rx + Tx 65-127 Octet Frames

The total number of frames of size 65 to 127 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 65 to 127 bytes long.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

#### 9.4.33.3.4 Rx + Tx 128-255 Octet Frames

The total number of frames of size 128 to 255 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 128 to 255 bytes long.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

For Rx reference only, see RFC1757 Ref. 1.13 etherStatsPkts128to255Octets.

#### 9.4.33.3.5 Rx + Tx 256-511 Octet Frames

The total number of frames of size 256 to 511 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 256 to 511 bytes long.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic. Rx + Tx 512-1023 Octet Frames

The total number of frames of size 512 to 1023 bytes received and transmitted on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 512 to 1023 bytes long.

CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.

#### 9.4.33.3.6 Rx + Tx 1024\_Up Octet Frames

The total number of frames of size 1024 to **pn\_rx\_maxlen** bytes for receive or 1024 up for transmit on the port. Such a frame is defined to be:

- any data or MAC control frame which was destined for any unicast, broadcast or multicast address, and
- did not experience late collisions, excessive collisions, or carrier sense error, and
- was 1024 to pn\_rx\_maxlen bytes long on receive, or any size on transmit.

CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.

**Table 9-29. Rx Statistics Summary**

Rx Stati- stic	Fra- me/ Oct	Rx/ Rx+ Tx	Frame Type					Frame Size (bytes)							Event					addr disc.
			MAC control		Data			< 64	64	65-127	128-255	256-511	512-1023	1024-rx_maxlen	> rx_maxlen	flow coll.	CRC error	align / code	over-run	
			Pause frame	Non-pause	Multicast	Broadcast	Unicast													
Good Rx Frames	F	Rx	(y	y	y	y	y	n	(y	y	y	y	y	y	n	-	n	n	-	n
Broadcast Rx Frames	F	Rx	(%	%	n	y)	n	n	(y	y	y	y	y	y	n	-	n	n	-	n

**Table 9-29. Rx Statistics Summary (continued)**

Multicast Rx Frames	F	Rx	(%)	%	y)	n	n	n	(y	y	y	y	y	y)	n	-	n	n	-	n
Pause Rx Frames	F	Rx	y	n	n	n	n	n	(y	y	y	y	y	y)	n	-	n	n	-	-
Rx CRC Errors	F	Rx	(y	y	y	y	y)	n	(y	y	y	y	y	y)	n	-	y	n	-	n
Rx Align/Code Errors	F	Rx	(y	y	y	y	y)	n	(y	y	y	y	y	y)	n	-	-	y	-	n
Over sized Rx Frames	F	Rx	(y	y	y	y	y)	n	n	n	n	n	n	n	y	-	n	n	-	n
Rx Jabbers	F	Rx	(y	y	y	y	y)	n	n	n	n	n	n	n	y	-	(y	y)	-	n
Undersized Rx Frames	F	Rx	n	n	(y	y	y)	y	n	n	n	n	n	n	n	-	n	n	-	n
Rx Fragments	F	Rx	n	n	(y	y	y)	y^	n	n	n	n	n	n	n	-	(y	y)	-	-
Rx Overruns	F	Rx	(y	y	y	y	y)	(y	y	y	y	y	y	y)	-	-	-	y	n	
64octet Frames	F	Rx+Tx	(y	y	y	y	y)	n	y	n	n	n	n	n	n	-	-	-	-	n
65-127octet Frames	F	Rx+Tx	(y	y	y	y	y)	n	n	y	n	n	n	n	n	-	-	-	-	n
128-255octet Frames	F	Rx+Tx	(y	y	y	y	y)	n	n	n	y	n	n	n	n	-	-	-	-	n

**Table 9-29. Rx Statistics Summary (continued)**

<b>256-511 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	n	n	y	n	n	n	-	-	-	-	n
<b>512-1023 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	n	n	n	y	n	n	-	-	-	-	n
<b>1024-UP octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	n	n	n	y	n	n	-	-	-	-	n
<b>Rx Octets</b>	O	Rx	(y	y	y	y	y)	n	(y	y	y	y	y	y)	n	-	n	n	-	n
<b>Net Octets</b>	O	Rx+Tx	(y	y	y	y	y)	(y	y	y	y	y	y	y	y)	-	-	-	-	-

1. “AND” is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning “OR” is indicated. Parentheses are significant.
2. “-” indicates conditions which are ignored in the formations of the statistic.
3. Statistics marked “Rx+Tx” are formed by summing the Rx and Tx statistics, each of which is formed independently.
4. The non-pause column refers to all MAC control frames (i.e. frames with length/type=88.08) with opcodes other than 0x0001. The pauseframe column refers to MAC frames with the opcode=0x0001.
5. The multicast, broadcast and unicast columns in the table refer to non-MAC Control/non-pause frames (i.e. data frames)
6. “%” If either a MAC control frame or pause frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
7. “%” If either a MAC control frame or pause frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
8. “y^” Frame fragments are not counted if less than 8 bytes.
9. flow coll. are half-duplex collisions forced by the MAC to achieve flow-control. A collision will be forced during the first 8 bytes so should not show in frame fragments. Some of the ‘-’s in this column might in reality be ‘n’s.
10. The rx\_overruns stat show above is for rx\_mof\_overruns and rx\_sof\_overruns added together.

**Table 9-30. Tx Statistics Summary**

Tx Statistic	Frame/Oct	Tx/Rx+Tx	Frame Type					Frame Size (bytes)							Event									
			MAC control		Data			64	65-127	128-255	256-511	512-1023	1024-1535	>1535	CRC error	Collision type					No carrier	Queued	Deferred	Underrun
			Pause (MAC)	Any (CPU)	Multicast	Broadcast	Unicast									flow	1	2-15	16	late				
<b>Good Tx Frames</b>	F	Tx	(y	y	y	y	y)	(y	y	y	y	y	y	y)	-	-	-	-	n	n	n	-	-	n

**Table 9-30. Tx Statistics Summary (continued)**

Bro ad c a s t T x F r a m e s	F	Tx	n	(%)	n	y	n	(y)	y	y	y	y	y	y	-	-	-	-	n	n	n	-	-	n
Mul ti ca s t T x F r a m e s	F	Tx	(y)	(%)	y	n	n	(y)	y	y	y	y	y	y	-	-	-	-	n	n	n	-	-	n
Pa u s e T x F r a m e s	F	Tx	y	n	n	n	n	y	n	n	n	n	n	n	-	-	-	-	-	-	-	-	-	-
Coll is io n s	F	Tx	n	(y)	y	y	y	(y)	y	y	y	y	y	y	-	(+	+	+	+	+	n	-	-	-
Sin g l e Coll is io n T x F r a m e s	F	Tx	n	(y)	y	y	y	(y)	y	y	y	y	y	y	-	-	y	n	n	n	n	-	-	-
Mul ti pl e Coll is io n T x F r a m e s	F	Tx	n	(y)	y	y	y	(y)	y	y	y	y	y	y	-	-	n	y	n	n	n	-	-	-
Exc ess i ve Coll is io n s	F	Tx	n	(y)	y	y	y	(y)	y	y	y	y	y	y	-	-	n	n	y	n	n	-	-	-
Lat e Coll is io n s	F	Tx	n	(y)	y	y	y	n	(y)	y	y	y	y	y	-	-	-	-	-	y	-	-	-	-
Def erre d T x F r a m e s	F	Tx	n	(y)	y	y	y	(y)	y	y	y	y	y	y	-	-	n	n	n	n	n	-	y	n

**Table 9-30. Tx Statistics Summary (continued)**

<b>Carrier Sense Errors</b>	F	Tx	(y	y	y	y	y)	(y	y	y	y	y	y	y)	-	-	-	-	-	-	y	-	-	-
<b>640 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	y	n	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
<b>65-127 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	y	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
<b>128-255 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	y	n	n	n	n	-	-	-	-	n	n	n	-	-	-
<b>256-511 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	n	y	n	n	n	-	-	-	-	n	n	n	-	-	-
<b>512-1023 octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	n	n	y	n	n	-	-	-	-	n	n	n	-	-	-
<b>1024-UP octet Frames</b>	F	Rx+Tx	(y	y	y	y	y)	n	n	n	n	n	y	y	-	-	-	-	n	n	n	-	-	-
<b>Tx Octets</b>	O	Tx	(y	y	y	y	y)	(y	y	y	y	y	y	y)	-	-	-	-	n	n	n	-	-	n
<b>Net Octets</b>	O	Rx+Tx	(y	y	y	y	y)	(y	y	y	y	y	y	y)	-	-	\$	\$	\$	\$	\$	-	-	-

1. "AND" is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning "OR" is indicated. Parentheses are significant.
2. "-" indicates conditions which are ignored in the formations of the statistic.
3. Statistics marked "Rx+Tx" are formed by summing the Rx and Tx statistics, each of which is formed independently.



4. Pause(MAC) frames are issued in the MAC as perfect (no CRC error) 64 byte frames in full duplex only, so cannot collide
5. “%” If a CPU sourced MAC control frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
6. “+” indicates collisions which are “summed” (i.e. every collision is counted in the Collisions statistic). Jam sequences used for half-duplex flow control are also counted.
7. “\$” Every byte written on the wire during each retry attempt is also counted in addition to frames which experience no collisions or carrier loss.
8. The flow collision type is for half-duplex collisions forced by the MAC to achieve flow control. Some of the ‘-’s in this column might in reality be ‘n’-s. To prevent double-counting, Net Octets are unaffected by the jam sequence – the ‘received’ bytes, however, are counted. (see Rx Statistics table).
9. When the transmit Tx FIFO is drained due to the MAC being disabled or link being lost, then the frames being purged will not appear in the Tx statistics.

#### 9.4.34 CPPI Streaming Packet Interface

The receive streaming interface on port 0 of the CPSW is responsible for receiving packet for Ethernet egress data from the packet streaming switch in the NAVSS. The CPPI receive port is equivalent to an Ethernet port with the difference being that the data is provided to the CPSW in the 128-bit streaming interface data format instead RGMII data format.

In addition to the packet data, the receive streaming interface also can provide additional control information that resides in the information words of the descriptor that was transmitted to the CPSW.

The tables below show the information that may be passed along with which descriptor information word to put it in.

##### 9.4.34.1 Port 0 CPPI Transmit Packet Streaming Interface (CPSW\_2G Egress)

INFO Word 0–3 and Status Data Word 0–3 (on EOP) are the only non-payload data word types that are transferred. Long packets are truncated at the CPWS0\_PN\_RX\_MAXLEN\_REG[13-0] RX\_MAXLEN byte value of the ingress port (only the CPWS0\_PN\_RX\_MAXLEN\_REG number of bytes are kept if long packets are transferred due to CPWS0\_PN\_MAC\_CONTROL\_REG register copy error frames set - RX\_CEF\_EN). MAC control frames are only transferred if the receiving Ethernet port has the CPWS0\_PN\_MAC\_CONTROL\_REG[24] RX\_CMF\_EN bit set.

INFO Words are a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

**Figure 9-13. TX INFO Word 0 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PKT_TYPE					RESERVED			PASS_CRC	CRC_T YPE	RESERVED					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FLOW_ID							

Bit	Field	Description
31-27	PKT_TYPE	Always set to 0b00111. Host PD (Packet Descriptor) Word 2. Packet Type: bits[31-27].
26-24	RESERVED	Reserved.
23	PASS_CRC	This bit is cleared to zero (no CRC passed) when the P0_TX_CRC_REMOVE bit in the CPSW_CONTROL_REG register is set (and the egress packet has no errors). When the remove bit is cleared to zero then this bit is cleared and no CRC is passed with the output packet. The packet length includes the CRC if it is present.

Bit	Field	Description
22	CRC_TYPE	The packet CRC type. The type of CRC passed is determined by CRC_TYPE field in the CPSW_PN_MAC_CONTROL_REG register (not by the type of CRC the packet had on Ethernet port ingress). Host PD Word 1. Protocol Specific Flags: bits[27-24]. 0h: Ethernet CRC 1h: Castagnoli CRC
21-8	RESERVED	Reserved.
7-0	FLOW_ID	This is the packet output transmit streaming interface flow. The default flow ID can be overridden by ALE classification (Thread mapping). The switch default flow is the 3-bit "From Port" value concatenated with the 3-bit "Switch Priority" {From_Port[2:0], Switch_Priority[2:0]} as shown below: Host PD (Packet Descriptor) Word 1. Flow ID: bits[13-0]. 0h: The packet was received on Ethernet port 1 1h: The packet was received on Ethernet port 2 Switch Priority – The actual hardware switch priority that the packet was stored in on the CPPI transmit FIFO.

**Figure 9-14. TX INFO Word 1 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0x4 (fixed_ps_size)												0			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0		PKT_LENGTH													

Bit	Field	Description
31-20	FIXED_PS_SIZE	Fixed ps size: 0x4
19-14	RESERVED	Reserved.
13-0 (Host PD (Packet Descriptor) Word 1. Packet Length: bits[ 21-0])	PKT_LENGTH	Specifies the number of bytes in the entire packet. Offset bytes are not included. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths.

**Figure 9-15. TX INFO Word 2 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0xFFFF															

**Figure 9-16. TX INFO Word 3 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0								SRC_ID							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															

Bit	Field	Description
31-24	RESERVED	Reserved.

Bit	Field	Description
23-16	SRC_ID	The packet SRC_ID value comes from the PORT1 field in the CPSW_P0_SRC_ID_A_REG register. (src_tag) PD (Packet Descriptor) Word 3. Source Tag Low bits[23-16] if RFLOW[a]_RFC.rx_src_tag_lo_sel = 0x4 or (src_tag) PD (Packet Descriptor) Word 3. Source Tag High bits[31-24] if RFLOW[a]_RFC.rx_src_tag_hi_sel = 0x4
15-0	RESERVED	Reserved.

### Note

TX Status Data Word [0..3] are mapped to Host Packet Descriptor Protocol Specific Words if RFLOW[a]\_RFA.rx\_psinfo\_present = 1 and RFLOW[a]\_RFA.rx\_ps\_location = 0

**Figure 9-17. TX Status Data Word 0 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMESTAMP[31:0]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP[31:0]															

Bit	Field	Description
31-0	TIMESTAMP[31:0]	Contains the lower 32-bits of the time stamp value.

**Figure 9-18. TX Status Data Word 1 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMESTAMP[63:32]															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMESTAMP[63:32]															

Bit	Field	Description
31-0	TIMESTAMP[63:32]	Contains the upper 32-bits of the time stamp value.

**Figure 9-19. TX Status Data Word 2 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED											IPV4_VALID	IPV6_VALID	TCP_UDP_N	FRAGMENT	CHECKSUM_ERROR
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHECKSUM_ADD															

Bit	Field	Description
31-21	RESERVED	Reserved.
20	IPV4_VALID	An IPV4 TCP or UDP Packet was detected.
19	IPV6_VALID	An IPV6 TCP or UDP Packet was detected.
18	TCP_UDP_N	Valid only when either the IPV4_VALID or IPV6_VALID bits are set. 0h: Indicates UDP packet was detected. 1h: Indicates TCP packet was detected.

Bit	Field	Description
17	FRAGMENT	Indicates that an IP fragment was detected. Valid only when either the IPV4_VALID or IPV6_VALID bits are set.
16	CHECKSUM_ERROR	Valid only when either the IPV4_VALID or IPV6_VALID bits are set.
15-0	CHECKSUM_ADD	This is the value that was summed during the checksum computation. This value is FFFFh for IPV4/6 UDP/TCP packets with no checksum error.

**Figure 9-20. TX Status Data Word 3 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0															

Bit	Field	Description
31-0	RESERVED	Reserved.

#### 9.4.34.2 CPPI Receive Packet Streaming Interface (CPSW Ingress)

Info Word 0/1/2/3 (INFO1 and INFO3 are ignored) are transferred on SOP. If a timestamp word (Extended Packet Info Word 0/1/2/3) is to be transferred it must be after SOP and before any packet data is transferred. Any following non-data type words will be dropped. The EOP word must be payload data.

Input receive packets cannot be aborted by the host. The INFO Word bit descriptions and Extended Packet INFO Word bit descriptions are shown below. The PASS\_CRC bit indicates that the CRC is passed with the packet data. Packets that have a passed CRC that is an error CRC will be output on the Ethernet port with at least one CRC byte inverted to indicate the error if P0\_RX\_PASS\_CRC\_ERR bit is set, otherwise they are dropped. The packet is a directed packet when any of the TO\_PORT bits are nonzero. A packet may be directed only to a single port. The packet will be sent to the port number indicated. For directed packets the lookup process is skipped to determine the destination. However, in vlan aware mode (when VLAN\_AWARE bit in the CPSW\_CONTROL\_REG register is set to 1h) the lookup is performed to determine untagged egress. Packets longer than the value in CPSW\_P0\_RX\_MAXLEN\_REG[13-0] RX\_MAXLEN bit field are dropped. Packets shorter than 60-Bytes are padded to 64-Bytes (after adding pad and CRC) if P0\_RX\_PAD bit in the CPSW\_CONTROL\_REG register is set and if PASS\_CRC is clear, otherwise they are dropped. This means that packets shorter than 64-Bytes are dropped if the PASS\_CRC info bit is set regardless of P0\_RX\_PAD bit (packets are padded only if they are short and do not have CRC).

A RX INFO word is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

#### Note

RX Control Data Words [0..2] are mapped to Host Packet Descriptor Protocol Specific Words if (TCHAN[a]\_TCFG.tx\_filt\_pswords = 0) and (Host PD Word 1.Protocol Specific Region Location.bit[28] = 0h) and (Host PD Word 1.Protocol Specific Valid Word Count.bits[22-27] = 4h)

**Figure 9-21. RX INFO Word 0 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED								PASS_CRC	CRC_T YPE	RESERVED					
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

Bit	Field	Description
31-24	RESERVED	Reserved.

Bit	Field	Description
23	PASS_CRC	The PASS_CRC bit indicates that the CRC is passed with the packet data. 0h: CRC is not passed with packet (CRC_TYPE is don't care) 1h: CRC of type CRC_TYPE is passed with the packet.
22 (Host PD (Packet Descriptor) Word 1. Protocol Specific Flags: bits[27-24])	CRC_TYPE	CRC Type 0h: Ethernet CRC 1h: Castagnoli CRC
21-0	RESERVED	Reserved.

**Figure 9-22. RX INFO Word 2 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED											TO_PORT				
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															

Bit	Field	Description
31-21	RESERVED	Reserved.
20-16 (Host PD (Packet Descriptor) Word 3. Dest Tag Low bits[8-0])	TO_PORT	Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode. 0h: Not directed 1h: Send the packet to port 1.
15-0	RESERVED	Reserved.

**Figure 9-23. RX Control Data Word 1 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
TIMES TAMP_ EN	RESERVED			DOMAIN								MSG_TYPE			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEQUENCE_ID															

Bit	Field	Description
31	TIMESTAMP_EN	When set, this bit indicates that the packet will generate a timesync event on Ethernet egress (if the CPTS is configured properly) with the associated DOMAIN, MSG_TYPE, and SEQUENCE_ID.
30-28	RESERVED	Reserved.
27-20	DOMAIN	Timesync domain.
19-16	MSG_TYPE	Timesync message type.
15-0	SEQUENCE_ID	Timesync sequence ID.

**Figure 9-24. RX Control Data Word 2 Format**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
----	----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

**Figure 9-24. RX Control Data Word 2 Format (continued)**

CHECKSUM_RESULT								CHECKSUM_START_BYTE							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CHECKSUM_INV	RESERVED	CHECKSUM_BYTECOUNT													

Bit	Field	Description
31-24	CHECKSUM_RESULT	This is the packet byte number where the checksum result will be placed in the egress packet. The first packet byte which is the first byte of the destination address is Byte 1 (not byte zero).
23-16	CHECKSUM_START_BYTE	This is the packet byte number to start the checksum calculation on. The first packet byte is Byte 1.
15	CHECKSUM_INV	When set, a zero checksum value will be inverted and sent as FFFFh.
14	RESERVED	Reserved.
13-0	CHECKSUM_BYTECOUNT	This is the number of bytes to calculate the checksum on. The outgoing Ethernet packet will have a checksum inserted when this value is non-zero.

Other INFO words are not taken into account.

#### 9.4.34.3 CPPI Checksum Offload

The CPPI host port can be enabled to perform checksum offload on host port packet ingress and egress. UDP (User Datagram Protocol) and TCP (Transmission Control Protocol) over IPV4 and IPV6 are supported. For the purposes of checksum description, the first packet byte (the first byte of the destination address) is byte 1 (not byte 0). That is, a 64 byte packet goes from byte 1 to byte 64. For all packet types, the S\_CN\_SWITCH bit in the CPSW\_CONTROL\_REG register must be set for the Outer VLAN L type to be supported.

##### 9.4.34.3.1 CPPI Transmit Checksum Offload

IPV4 and IPV6 UDP and TCP packets that are received on any Ethernet port and destined for port 0 egress are checked for correct checksum as described below. The byte counts below are shown for packets with no VLAN's. The byte counts vary with one or two packet VLAN's. Packets received on an Ethernet port with errors are not checked for a correct checksum if they are passed to the host.

##### 9.4.34.3.1.1 IPV4 UDP

- Byte 15 Upper Nibble = 4 for IPV4
- Byte 15 Lower Nibble = IHL - Nibble with number of 32-bit words in IPV4 header (5 to 15 supported).
- Bytes 20-21 = fragment[15-0] – Bit 13 is the MF bit and bits [12-0] are the Fragment offset. A packet is a fragment if the MF bit is set or if the fragment offset is non-zero. The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets have MF=0 and a zero offset. A count is output for packet fragments but no errors are reported. First fragments have the UDP header included in the count. Middle and last fragments have only data included in the count (there is no UDP header).
- Byte 24 = 0x11 for UDP protocol.
- Received packet UDP checksum of zero means that there is no IPV4 checksum sent with the packet so no error will be issued.
- Received packet UDP checksum of 0xFFFF means that the checksum was calculated to be 0xFFFF or 0x0000 but was sent in the transmitted packet as 0xFFFF by the sending originating entity.

##### 9.4.34.3.1.2 IPV4 TCP

- Byte 15 Upper Nibble = 4 for IPV4
- Byte 15 Lower Nibble = IHL - Nibble with number of 32-bit words in IPV4 header (5 to 15 supported).
- Bytes 20-21 = fragment[15-0] – Bit 13 is the MF bit and bits [12-0] are the Fragment offset. A packet is a fragment if the MF bit is set or if the fragment offset is non-zero. The first packet fragment has MF=1 with a

zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets have MF=0 and a zero offset. A count is output for packet fragments but no errors are reported. First fragments have the UDP header included in the count. Middle and last fragments have only data included in the count (there is no TCP header).

- Byte 24 = 0x06 for TCP protocol.

#### 9.4.34.3.1.3 IPV6 UDP

- Byte 15 upper nibble = 6 for IPV6.
- Byte 21 = 0x11 for UDP protocol as next header.
- Fragment extension headers are supported. First fragments have a fragment extension header (byte 21 = 0x2C) followed by a UDP header (byte 55 = 0x11). Middle and last fragments have a fragment extension header followed by data only (no UDP header). The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets do not have a fragment extension header. A count is output for packet fragments but no errors are reported.
- Received packet UDP checksum of zero means that there is no IPV6 checksum sent with the packet so no error will be issued.
- Received packet UDP checksum of 0xFFFF means that the checksum was calculated to be 0xFFFF or 0x0000 but was sent in the transmitted packet as 0xFFFF by the sending originating entity.

#### 9.4.34.3.1.4 IPV6 TCP

- Byte 15 upper nibble = 6 for IPV6.
- Byte 21 = 0x06 for TCP protocol as next header.
- Fragment extension headers are supported. First fragments have a fragment extension header (byte 21 = 0x2C) followed by a UDP header (byte 55 = 0x06). Middle and last fragments have a fragment extension header followed by data only (no TCP header). The first packet fragment has MF=1 with a zero offset. Middle fragments have MF=1 with a nonzero offset. The last packet fragment has MF=0 with a nonzero offset. Non-fragmented packets do not have a fragment extension header. A count is output for packet fragments but no errors are reported.

#### 9.4.34.4 CPPI Receive Checksum Offload

Packets sent from host port 0 (switch ingress) to any Ethernet port can have a checksum calculated and inserted into the Ethernet egress packet. The RX\_CHECKSUM\_EN bit in the CPSW\_P0\_CONTROL\_REG register must be set for receive checksum operation to be enabled. When bit RX\_CHECKSUM\_EN is enabled, Control Data Word 2 input on CPPI receive PSI interface determines how the checksum is calculated. The CHECKSUM\_RESULT field in Control Data Word 2 determines where the checksum is inserted. The checksum result location is adjusted by the egress port if a VLAN is to be inserted or removed on Ethernet port egress.

#### 9.4.35 Egress Packet Operations

##### Note

InterVLAN routing is supported on SR2.0 only.

Each CPSW egress port (Ethernet and Host) is capable of performing egress packet processing operations. IntraVLAN processing either adds, removes, or replaces VLAN information or does nothing. InterVLAN routing allows hardware routing between a limited number of VLANs - thereby allowing high-bandwidth or other routing operations to be offloaded from software to the CPSW (hardware). IntraVLAN processing and InterVLAN routing operations are mutually exclusive. In addition, the packet source and destination addresses can be swapped on egress to facilitate OAM or generic testing operations.

## 9.5 MCU\_CPSW0 Programming Guide

### 9.5.1 Initialization and Configuration of CPSW Subsystem

To configure the CPSW Ethernet Subsystem for operation, the host must perform the following:



1. Select the Interface (RMII, or RGMII ) Mode. See the CTRLMMR\_MCU\_ENET\_CTRL[1-0] MODE\_SEL register.
2. Configure pads (pin muxing), as per the interface selected. Refer to *Pad Configuration Registers* and the device-specific Datasheet.
3. Enable the CPSW Ethernet Subsystem clocks. See [Section 9.3, CPSW Integration](#)
4. Ensure that at least 2000 CPPI\_ICLK periods are run after reset is de-asserted.
5. Configure the CPSW\_CONTROL\_REG register
6. Configure the Ethernet Port Source Address registers (CPSW0\_PN\_SA\_L\_REG and CPSW\_PN\_SA\_H\_REG)
7. Configure the CPSW statistic port enable register CPSW\_STAT\_PORT\_EN\_REG
8. Configure the ALE ([Section 9.4.7.1, Address Lookup Engine](#))
9. Configure the MDIO ([Section 9.5.3.1, Initializing the MDIO Module](#))
10. Configure Ethernet port, as per the desired mode of operations

### 9.5.2 CPSW Reset

To reset the CPMAC\_SL, the host must perform the following:

1. Set CMD\_IDLE in the Ethernet port CPSW\_PN\_MAC\_CONTROL\_REG register
2. Wait for IDLE to be indicated in the Ethernet port CPSW\_PN\_MAC\_STATUS\_REG register
3. Configure the Ethernet ports.

### 9.5.3 MDIO Software Interface

#### 9.5.3.1 Initializing the MDIO Module

The following steps are performed by the application software or device driver to initialize the MDIO device:

1. Configure the PREAMBLE and CLKDIV bits in the MDIO Control register (CPSW\_MDIO\_CONTROL\_REG).
2. Enable the MDIO module by setting the ENABLE bit in CPSW\_MDIO\_CONTROL\_REG.
3. The MDIO PHY alive status register (MDIO CPSW\_MDIO\_ALIVE\_REG) can be read in polling fashion until a PHY connected to the system responded, and the MDIO PHY link status register (MDIO CPSW\_MDIO\_LINK\_REG) can determine whether this PHY already has a link.
4. Set the appropriate PHY addresses in the MDIO user PHY select register (CPSW\_MDIO\_USER\_PHY\_SEL\_REG\_k, where k = 0 or 1), and set the LINKINT\_ENABLE bit to enable a link change event interrupt if desirable.
5. Set the appropriate LINKSEL bit in the CPSW\_MDIO\_USER\_PHY\_SEL\_REG\_k register (where k = 0 or 1).
6. Set the appropriate USERINTMASKSET bit field in the CPSW\_MDIO\_USER\_INT\_MASK\_SET\_REG register.
7. If an interrupt on general MDIO register access is desired, set the corresponding bit in the MDIO user command complete interrupt mask set register (MDIO CPSW\_MDIO\_USER\_INT\_MASK\_SET\_REG) to use the MDIO user access register (MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k, where k = 0 or 1).

#### 9.5.3.2 Writing Data To a PHY Register

The MDIO module includes a user access register (MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k, where k = 0 or 1) to directly access a specified PHY device. To write a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k) is cleared.
2. Write to the GO, WRITE, REGADR, PHYADR, and DATA bits in MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k corresponding to the PHY and PHY register SW wants to write.
3. The write operation to the PHY is scheduled and completed by the MDIO module. Completion of the write operation can be determined by polling the GO bit in MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k for a 0.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (CPSW\_MDIO\_USER\_INT\_RAW\_REG) corresponding to MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (CPSW\_MDIO\_USER\_INT\_MASK\_SET\_REG),



then the bit is also set in the MDIO user command complete interrupt register (CPSW\_MDIO\_USER\_INT\_MASKED\_REG ) and an interrupt is triggered on the host processor.

#### 9.5.3.3 Reading Data From a PHY Register

The MDIO module includes a user access register (MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k, where k = 0 or 1) to directly access a specified PHY device. To read a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (CPSW\_MDIO\_USER\_ACCESS\_REG\_k, where k = 0 or 1) is cleared.
2. Write to the GO, REGADR, and PHYADR bits in the CPSW\_MDIO\_USER\_ACCESS\_REG\_k register corresponding to the PHY and PHY register SW wants to read.
3. The read data value is available in the DATA bit field in MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k register after the module completes the read operation on the serial bus. Completion of the read operation can be determined by polling the GO and ACK bits in CPSW\_MDIO\_USER\_ACCESS\_REG\_k register. After the GO bit has cleared, the ACK bit is set on a successful read.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (CPSW\_MDIO\_USER\_INT\_RAW\_REG) corresponding to MDIO CPSW\_MDIO\_USER\_ACCESS\_REG\_k used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (CPSW\_MDIO\_USER\_INT\_MASK\_SET\_REG), then the bit is also set in the MDIO user command complete interrupt register (CPSW\_MDIO\_USER\_INT\_MASKED\_REG) and an interrupt is triggered on the host processor.

## 9.6 MSS\_CPSW Registers

**Table 9-31** lists the memory-mapped registers for the MSS\_CPSW registers. All register offset addresses not listed in **Table 9-31** should be considered as reserved locations and the register contents should not be modified.

**Table 9-31. MSS\_CPSW Registers**

Offset	Acronym	Register Name	Section
0h	CPSW_NUSS_IDVER_REG	ID Version Register	<a href="#">Go</a>
4h	SS_SYNCE_COUNT_REG	SyncE Count Register	<a href="#">Go</a>
8h	SS_SYNCE_MUX_REG	SyncE Mux Register	<a href="#">Go</a>
Ch	SS_CONTROL_REG	Control Register	<a href="#">Go</a>
18h	SS_INT_CONTROL_REG	Interrupt Control Register	<a href="#">Go</a>
1Ch	SS_STATUS_REG	Subsystem Status Register	<a href="#">Go</a>
20h	SUBSYSTEM_CONFIG_REG	Subsystem Configuration Register	<a href="#">Go</a>
30h	RGMII1_STATUS_REG	RGMII1 Status Register	<a href="#">Go</a>
F00h	MDIO_MDIO_VERSION_REG	version_reg	<a href="#">Go</a>
F04h	MDIO_CONTROL_REG	control_reg	<a href="#">Go</a>
F08h	MDIO_ALIVE_REG	alive_reg	<a href="#">Go</a>
F0Ch	MDIO_LINK_REG	link_reg	<a href="#">Go</a>
F10h	MDIO_LINK_INT_RAW_REG	link_int_raw_reg	<a href="#">Go</a>
F14h	MDIO_LINK_INT_MASKED_REG	link_int_masked_reg	<a href="#">Go</a>
F18h	MDIO_LINK_INT_MASK_SET_REG	link_int_mask_set_reg	<a href="#">Go</a>
F1Ch	MDIO_LINK_INT_MASK_CLEAR_REG	link_int_mask_clear_reg	<a href="#">Go</a>
F20h	MDIO_USER_INT_RAW_REG	user_int_raw_reg	<a href="#">Go</a>
F24h	MDIO_USER_INT_MASKED_REG	user_int_masked_reg	<a href="#">Go</a>
F28h	MDIO_USER_INT_MASK_SET_REG	user_int_mask_set_reg	<a href="#">Go</a>
F2Ch	MDIO_USER_INT_MASK_CLEAR_REG	user_int_mask_clear_reg	<a href="#">Go</a>
F30h	MDIO_MANUAL_IF_REG	manual_if_reg	<a href="#">Go</a>
F34h	MDIO_POLL_REG	poll_reg	<a href="#">Go</a>
F38h	MDIO_POLL_EN_REG	poll_reg	<a href="#">Go</a>
F3Ch	MDIO_CLAUS45_REG	poll_reg	<a href="#">Go</a>
F40h	MDIO_USER_ADDR0_REG	poll_reg	<a href="#">Go</a>
F44h	MDIO_USER_ADDR1_REG	poll_reg	<a href="#">Go</a>
F80h	USER_GROUP0_USER_ACCESS_REG	user_access_reg	<a href="#">Go</a>
F84h	USER_GROUP0_USER_PHY_SEL_REG	user_phy_sel_reg	<a href="#">Go</a>
F88h	USER_GROUP1_USER_ACCESS_REG	user_access_reg	<a href="#">Go</a>
F8Ch	USER_GROUP1_USER_PHY_SEL_REG	user_phy_sel_reg	<a href="#">Go</a>
1800h	REGS_INT_SS_C0_TH_THRESH_PULSE_EN_REG	Core 0 THost Threshold Pulse Interrupt Enable Register	<a href="#">Go</a>
1804h	REGS_INT_SS_C0_TH_PULSE_EN_REG	Core 0 THost Pulse Interrupt Enable Register	<a href="#">Go</a>
1808h	REGS_INT_SS_C0_FH_PULSE_EN_REG	Core 0 FHost Pulse Interrupt Enable Register	<a href="#">Go</a>
180Ch	REGS_INT_SS_C0_MISC_EN_REG	Core 0 Misc Interrupt Enable Register	<a href="#">Go</a>
1810h	REGS_INT_SS_C0_TH_THRESH_PULSE_STATUS_REG	Core 0 THost Threshold Pulse Interrupt Status Register	<a href="#">Go</a>
1814h	REGS_INT_SS_C0_TH_PULSE_STATUS_REG	Core 0 THost Pulse Interrupt Status Register	<a href="#">Go</a>
1818h	REGS_INT_SS_C0_FH_PULSE_STATUS_REG	Core 0 FHost Pulse Interrupt Status Register	<a href="#">Go</a>
181Ch	REGS_INT_SS_C0_MISC_STATUS_REG	Core 0 Misc Interrupt Status Register	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
1820h	REGS_INT_SS_C0_TH_IMAX_REG	Core 0 THost Interrupt Max Register Register	<a href="#">Go</a>
1824h	REGS_INT_SS_C0_FH_IMAX_REG	Core 0 FHost Interrupt Max Register Register	<a href="#">Go</a>
00020000h	CPSW_NC_CPSW_ID_VER_REG	idver_reg	<a href="#">Go</a>
00020004h	CPSW_NC_CONTROL_REG	control_reg	<a href="#">Go</a>
00020010h	CPSW_NC_EM_CONTROL_REG	em_control_reg	<a href="#">Go</a>
00020014h	CPSW_NC_STAT_PORT_EN_REG	stat_port_en_reg	<a href="#">Go</a>
00020018h	CPSW_NC_PTYPE_REG	ptype_reg	<a href="#">Go</a>
0002001Ch	CPSW_NC_SOFT_IDLE_REG	soft_idle_reg	<a href="#">Go</a>
00020020h	CPSW_NC_THRU_RATE_REG	thru_rate_reg	<a href="#">Go</a>
00020024h	CPSW_NC_GAP_THRESH_REG	gap_thresh_reg	<a href="#">Go</a>
0002002Ch	CPSW_NC_EEE_PRESCALE_REG	eee_prescale_reg	<a href="#">Go</a>
00020030h	CPSW_NC_TX_G_OFLOW_THRESH_SE T_REG	tx_g_oflow_thresh_set_reg	<a href="#">Go</a>
00020034h	CPSW_NC_TX_G_OFLOW_THRESH_CL R_REG	tx_g_oflow_thresh_clr_reg	<a href="#">Go</a>
00020038h	CPSW_NC_TX_G_BUF_THRESH_SET_L _REG	tx_g_buf_thresh_set_l_reg	<a href="#">Go</a>
0002003Ch	CPSW_NC_TX_G_BUF_THRESH_SET_H _REG	tx_g_buf_thresh_set_h_reg	<a href="#">Go</a>
00020040h	CPSW_NC_TX_G_BUF_THRESH_CLR_L _REG	tx_g_buf_thresh_clr_l_reg	<a href="#">Go</a>
00020044h	CPSW_NC_TX_G_BUF_THRESH_CLR_ H_REG	tx_g_buf_thresh_clr_h_reg	<a href="#">Go</a>
00020050h	CPSW_NC_VLAN_LTYPE_REG	vlan_ltype_reg	<a href="#">Go</a>
00020054h	CPSW_NC_EST_TS_DOMAIN_REG	est_ts_domain_reg	<a href="#">Go</a>
00020100h	CPSW_NC_TX_PRI0_MAXLEN_REG	tx_pri0_maxlen_reg	<a href="#">Go</a>
00020104h	CPSW_NC_TX_PRI1_MAXLEN_REG	tx_pri1_maxlen_reg	<a href="#">Go</a>
00020108h	CPSW_NC_TX_PRI2_MAXLEN_REG	tx_pri2_maxlen_reg	<a href="#">Go</a>
0002010Ch	CPSW_NC_TX_PRI3_MAXLEN_REG	tx_pri3_maxlen_reg	<a href="#">Go</a>
00020110h	CPSW_NC_TX_PRI4_MAXLEN_REG	tx_pri4_maxlen_reg	<a href="#">Go</a>
00020114h	CPSW_NC_TX_PRI5_MAXLEN_REG	tx_pri5_maxlen_reg	<a href="#">Go</a>
00020118h	CPSW_NC_TX_PRI6_MAXLEN_REG	tx_pri6_maxlen_reg	<a href="#">Go</a>
0002011Ch	CPSW_NC_TX_PRI7_MAXLEN_REG	tx_pri7_maxlen_reg	<a href="#">Go</a>
00021004h	CPSW_NC_CPPI_P0_CONTROL_REG	p0_control_reg	<a href="#">Go</a>
00021008h	CPSW_NC_CPPI_P0_FLOW_ID_OFFSET _REG	p0_flow_id_offset_reg	<a href="#">Go</a>
00021010h	CPSW_NC_CPPI_P0_BLK_CNT_REG	p0_blk_cnt_reg	<a href="#">Go</a>
00021014h	CPSW_NC_CPPI_P0_PORT_VLAN_REG	p0_port_vlan_reg	<a href="#">Go</a>
00021018h	CPSW_NC_CPPI_P0_TX_PRI_MAP_REG	p0_tx_pri_map_reg	<a href="#">Go</a>
0002101Ch	CPSW_NC_CPPI_P0_PRI_CTL_REG	p0_pri_ctl_reg	<a href="#">Go</a>
00021020h	CPSW_NC_CPPI_P0_RX_PRI_MAP_RE G	p0_rx_pri_map_reg	<a href="#">Go</a>
00021024h	CPSW_NC_CPPI_P0_RX_MAXLEN_REG	p0_rx_maxlen_reg	<a href="#">Go</a>
00021028h	CPSW_NC_CPPI_P0_TX_BLKs_PRI_RE G	p0_tx_blks_pri_reg	<a href="#">Go</a>
00021030h	CPSW_NC_CPPI_P0_IDLE2LPI_REG	p0_idle2lpi_reg	<a href="#">Go</a>
00021034h	CPSW_NC_CPPI_P0_LPI2WAKE_REG	p0_lpi2wake_reg	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00021038h	CPSW_NC_CPPI_P0_EEE_STATUS_REG	p0_eee_status_reg	<a href="#">Go</a>
0002103Ch	CPSW_NC_CPPI_P0_RX_PKTS_PRI_REG	p0_rx_pkts_pri_reg	<a href="#">Go</a>
0002104Ch	CPSW_NC_CPPI_P0_RX_GAP_REG	p0_rx_gap_reg	<a href="#">Go</a>
00021050h	CPSW_NC_CPPI_P0_FIFO_STATUS_REG	p0_fifo_status_reg	<a href="#">Go</a>
00021080h	CPSW_NC_CPPI_P0_MAX_BKTS_REG	p0_max_bkts_reg	<a href="#">Go</a>
00021120h	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_0	p0_rx_dscp_map_reg	<a href="#">Go</a>
00021124h	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_1	p0_rx_dscp_map_reg	<a href="#">Go</a>
00021128h	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_2	p0_rx_dscp_map_reg	<a href="#">Go</a>
0002112Ch	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_3	p0_rx_dscp_map_reg	<a href="#">Go</a>
00021130h	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_4	p0_rx_dscp_map_reg	<a href="#">Go</a>
00021134h	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_5	p0_rx_dscp_map_reg	<a href="#">Go</a>
00021138h	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_6	p0_rx_dscp_map_reg	<a href="#">Go</a>
0002113Ch	CPSW_NC_CPPI_P0_RX_DSCP_MAP_REG_7	p0_rx_dscp_map_reg	<a href="#">Go</a>
00021140h	CPSW_NC_CPPI_P0_PRI_CIR_REG_0	p0_pri_cir_reg	<a href="#">Go</a>
00021144h	CPSW_NC_CPPI_P0_PRI_CIR_REG_1	p0_pri_cir_reg	<a href="#">Go</a>
00021148h	CPSW_NC_CPPI_P0_PRI_CIR_REG_2	p0_pri_cir_reg	<a href="#">Go</a>
0002114Ch	CPSW_NC_CPPI_P0_PRI_CIR_REG_3	p0_pri_cir_reg	<a href="#">Go</a>
00021150h	CPSW_NC_CPPI_P0_PRI_CIR_REG_4	p0_pri_cir_reg	<a href="#">Go</a>
00021154h	CPSW_NC_CPPI_P0_PRI_CIR_REG_5	p0_pri_cir_reg	<a href="#">Go</a>
00021158h	CPSW_NC_CPPI_P0_PRI_CIR_REG_6	p0_pri_cir_reg	<a href="#">Go</a>
0002115Ch	CPSW_NC_CPPI_P0_PRI_CIR_REG_7	p0_pri_cir_reg	<a href="#">Go</a>
00021160h	CPSW_NC_CPPI_P0_PRI_EIR_REG_0	p0_pri_eir_reg	<a href="#">Go</a>
00021164h	CPSW_NC_CPPI_P0_PRI_EIR_REG_1	p0_pri_eir_reg	<a href="#">Go</a>
00021168h	CPSW_NC_CPPI_P0_PRI_EIR_REG_2	p0_pri_eir_reg	<a href="#">Go</a>
0002116Ch	CPSW_NC_CPPI_P0_PRI_EIR_REG_3	p0_pri_eir_reg	<a href="#">Go</a>
00021170h	CPSW_NC_CPPI_P0_PRI_EIR_REG_4	p0_pri_eir_reg	<a href="#">Go</a>
00021174h	CPSW_NC_CPPI_P0_PRI_EIR_REG_5	p0_pri_eir_reg	<a href="#">Go</a>
00021178h	CPSW_NC_CPPI_P0_PRI_EIR_REG_6	p0_pri_eir_reg	<a href="#">Go</a>
0002117Ch	CPSW_NC_CPPI_P0_PRI_EIR_REG_7	p0_pri_eir_reg	<a href="#">Go</a>
00021180h	CPSW_NC_CPPI_P0_TX_D_THRESH_SET_L_REG	p0_tx_d_thresh_set_l_reg	<a href="#">Go</a>
00021184h	CPSW_NC_CPPI_P0_TX_D_THRESH_SET_H_REG	p0_tx_d_thresh_set_h_reg	<a href="#">Go</a>
00021188h	CPSW_NC_CPPI_P0_TX_D_THRESH_CLR_L_REG	p0_tx_d_thresh_clr_l_reg	<a href="#">Go</a>
0002118Ch	CPSW_NC_CPPI_P0_TX_D_THRESH_CLR_H_REG	p0_tx_d_thresh_clr_h_reg	<a href="#">Go</a>
00021190h	CPSW_NC_CPPI_P0_TX_G_BUF_THRESH_SET_L_REG	p0_tx_g_buf_thresh_set_l_reg	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00021194h	CPSW_NC_CPPI_P0_TX_G_BUF_THRE SH_SET_H_REG	p0_tx_g_buf_thresh_set_h_reg	<a href="#">Go</a>
00021198h	CPSW_NC_CPPI_P0_TX_G_BUF_THRE SH_CLR_L_REG	p0_tx_g_buf_thresh_clr_l_reg	<a href="#">Go</a>
0002119Ch	CPSW_NC_CPPI_P0_TX_G_BUF_THRE SH_CLR_H_REG	p0_tx_g_buf_thresh_clr_h_reg	<a href="#">Go</a>
00021300h	CPSW_NC_CPPI_P0_SRC_ID_A_REG	p0_src_id_a_reg	<a href="#">Go</a>
00021304h	CPSW_NC_CPPI_P0_SRC_ID_B_REG	p0_src_id_b_reg	<a href="#">Go</a>
00021320h	CPSW_NC_CPPI_P0_HOST_BLKs_PRI_ REG	p0_host_blks_pri_reg	<a href="#">Go</a>
00022000h	CPSW_NC_ETH_MAC_0_PN_RESERVE D_REG	pn_reserved_reg	
00022004h	CPSW_NC_ETH_MAC_0_PN_CONTROL _REG	pn_control_reg	<a href="#">Go</a>
00022008h	CPSW_NC_ETH_MAC_0_PN_MAX_BLK S_REG	pn_max_blks_reg	<a href="#">Go</a>
00022010h	CPSW_NC_ETH_MAC_0_PN_BLK_CNT_ REG	pn_blk_cnt_reg	<a href="#">Go</a>
00022014h	CPSW_NC_ETH_MAC_0_PN_PORT_VLA N_REG	pn_port_vlan_reg	<a href="#">Go</a>
00022018h	CPSW_NC_ETH_MAC_0_PN_TX_PRI_M AP_REG	pn_tx_pri_map_reg	<a href="#">Go</a>
0002201Ch	CPSW_NC_ETH_MAC_0_PN_PRI_CTL_ REG	pn_pri_ctl_reg	<a href="#">Go</a>
00022020h	CPSW_NC_ETH_MAC_0_PN_RX_PRI_M AP_REG	pn_rx_pri_map_reg	<a href="#">Go</a>
00022024h	CPSW_NC_ETH_MAC_0_PN_RX_MAXL EN_REG	pn_rx_maxlen_reg	<a href="#">Go</a>
00022028h	CPSW_NC_ETH_MAC_0_PN_TX_BLKs_ PRI_REG	pn_tx_blks_pri_reg	<a href="#">Go</a>
0002202Ch	CPSW_NC_ETH_MAC_0_PN_RX_FLOW _THRESH_REG	pn_rx_flow_thresh_reg	<a href="#">Go</a>
00022030h	CPSW_NC_ETH_MAC_0_PN_IDLE2LPI_ REG	pn_idle2lpi_reg	<a href="#">Go</a>
00022034h	CPSW_NC_ETH_MAC_0_PN_LPI2WAKE _REG	pn_lpi2wake_reg	<a href="#">Go</a>
00022038h	CPSW_NC_ETH_MAC_0_PN_EEE_STAT US_REG	pn_eee_status_reg	<a href="#">Go</a>
00022050h	CPSW_NC_ETH_MAC_0_PN_FIFO_STA TUS_REG	pn_fifo_status_reg	<a href="#">Go</a>
00022060h	CPSW_NC_ETH_MAC_0_PN_EST_CON TROL_REG	pn_est_control_reg	<a href="#">Go</a>
00022120h	CPSW_NC_ETH_MAC_0_PN_RX_DSCP _MAP_REG_0	pn_rx_dscp_map_reg	<a href="#">Go</a>
00022124h	CPSW_NC_ETH_MAC_0_PN_RX_DSCP _MAP_REG_1	pn_rx_dscp_map_reg	<a href="#">Go</a>
00022128h	CPSW_NC_ETH_MAC_0_PN_RX_DSCP _MAP_REG_2	pn_rx_dscp_map_reg	<a href="#">Go</a>
0002212Ch	CPSW_NC_ETH_MAC_0_PN_RX_DSCP _MAP_REG_3	pn_rx_dscp_map_reg	<a href="#">Go</a>
00022130h	CPSW_NC_ETH_MAC_0_PN_RX_DSCP _MAP_REG_4	pn_rx_dscp_map_reg	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00022134h	CPSW_NC_ETH_MAC_0_PN_RX_DSCP_MAP_REG_5	pn_rx_dscp_map_reg	<a href="#">Go</a>
00022138h	CPSW_NC_ETH_MAC_0_PN_RX_DSCP_MAP_REG_6	pn_rx_dscp_map_reg	<a href="#">Go</a>
0002213Ch	CPSW_NC_ETH_MAC_0_PN_RX_DSCP_MAP_REG_7	pn_rx_dscp_map_reg	<a href="#">Go</a>
00022140h	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_0	pn_pri_send_reg	<a href="#">Go</a>
00022144h	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_1	pn_pri_send_reg	<a href="#">Go</a>
00022148h	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_2	pn_pri_send_reg	<a href="#">Go</a>
0002214Ch	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_3	pn_pri_send_reg	<a href="#">Go</a>
00022150h	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_4	pn_pri_send_reg	<a href="#">Go</a>
00022154h	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_5	pn_pri_send_reg	<a href="#">Go</a>
00022158h	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_6	pn_pri_send_reg	<a href="#">Go</a>
0002215Ch	CPSW_NC_ETH_MAC_0_PN_PRI_CIR_R EG_7	pn_pri_send_reg	<a href="#">Go</a>
00022160h	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_0	pn_pri_idle_reg	<a href="#">Go</a>
00022164h	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_1	pn_pri_idle_reg	<a href="#">Go</a>
00022168h	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_2	pn_pri_idle_reg	<a href="#">Go</a>
0002216Ch	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_3	pn_pri_idle_reg	<a href="#">Go</a>
00022170h	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_4	pn_pri_idle_reg	<a href="#">Go</a>
00022174h	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_5	pn_pri_idle_reg	<a href="#">Go</a>
00022178h	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_6	pn_pri_idle_reg	<a href="#">Go</a>
0002217Ch	CPSW_NC_ETH_MAC_0_PN_PRI_EIR_R EG_7	pn_pri_idle_reg	<a href="#">Go</a>
00022180h	CPSW_NC_ETH_MAC_0_PN_TX_D_THR ESH_SET_L_REG	pn_tx_d_thresh_set_l_reg	<a href="#">Go</a>
00022184h	CPSW_NC_ETH_MAC_0_PN_TX_D_THR ESH_SET_H_REG	pn_tx_d_thresh_set_h_reg	<a href="#">Go</a>
00022188h	CPSW_NC_ETH_MAC_0_PN_TX_D_THR ESH_CLR_L_REG	pn_tx_d_thresh_clr_l_reg	<a href="#">Go</a>
0002218Ch	CPSW_NC_ETH_MAC_0_PN_TX_D_THR ESH_CLR_H_REG	pn_tx_d_thresh_clr_h_reg	<a href="#">Go</a>
00022190h	CPSW_NC_ETH_MAC_0_PN_TX_G_BUF_THRESH_SET_L_REG	pn_tx_g_buf_thresh_set_l_reg	<a href="#">Go</a>
00022194h	CPSW_NC_ETH_MAC_0_PN_TX_G_BUF_THRESH_SET_H_REG	pn_tx_g_buf_thresh_set_h_reg	<a href="#">Go</a>
00022198h	CPSW_NC_ETH_MAC_0_PN_TX_G_BUF_THRESH_CLR_L_REG	pn_tx_g_buf_thresh_clr_l_reg	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0002219Ch	CPSW_NC_ETH_MAC_0_PN_TX_G_BUF_THRESH_CLR_H_REG	pn_tx_g_buf_thresh_clr_h_reg	<a href="#">Go</a>
00022300h	CPSW_NC_ETH_MAC_0_PN_TX_D_OFLOW_ADDVAL_L_REG	pn_tx_d_oflow_addval_l_reg	<a href="#">Go</a>
00022304h	CPSW_NC_ETH_MAC_0_PN_TX_D_OFLOW_ADDVAL_H_REG	pn_tx_d_oflow_addval_h_reg	<a href="#">Go</a>
00022308h	CPSW_NC_ETH_MAC_0_PN_SA_L_REG	pn_sa_l_reg	<a href="#">Go</a>
0002230Ch	CPSW_NC_ETH_MAC_0_PN_SA_H_REG	pn_sa_h_reg	<a href="#">Go</a>
00022310h	CPSW_NC_ETH_MAC_0_PN_TS_CTL_REG	pn_ts_ctl_reg	<a href="#">Go</a>
00022314h	CPSW_NC_ETH_MAC_0_PN_TS_SEQ_LTYPE_REG	pn_ts_seq_ltype_reg	<a href="#">Go</a>
00022318h	CPSW_NC_ETH_MAC_0_PN_TS_VLAN_LTYPE_REG	pn_ts_vlan_ltype_reg	<a href="#">Go</a>
0002231Ch	CPSW_NC_ETH_MAC_0_PN_TS_CTL_LTYPE2_REG	pn_ts_ctl_ltype2_reg	<a href="#">Go</a>
00022320h	CPSW_NC_ETH_MAC_0_PN_TS_CTL2_REG	pn_ts_ctl2_reg	<a href="#">Go</a>
00022330h	CPSW_NC_ETH_MAC_0_PN_MAC_CONTROL_REG	pn_mac_control_reg	<a href="#">Go</a>
00022334h	CPSW_NC_ETH_MAC_0_PN_MAC_STATUS_REG	pn_mac_status_reg	<a href="#">Go</a>
00022338h	CPSW_NC_ETH_MAC_0_PN_MAC_SOFT_RESET_REG	pn_mac_soft_reset_reg	<a href="#">Go</a>
0002233Ch	CPSW_NC_ETH_MAC_0_PN_MAC_BOFFTEST_REG	pn_mac_bofftest_reg	<a href="#">Go</a>
00022340h	CPSW_NC_ETH_MAC_0_PN_MAC_RX_PAUSETIMER_REG	pn_mac_rx_pausetimer_reg	<a href="#">Go</a>
00022350h	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_0	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
00022354h	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_1	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
00022358h	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_2	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
0002235Ch	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_3	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
00022360h	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_4	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
00022364h	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_5	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
00022368h	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_6	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
0002236Ch	CPSW_NC_ETH_MAC_0_PN_MAC_RXN_PAUSETIMER_REG_7	pn_mac_rxn_pausetimer_reg	<a href="#">Go</a>
00022370h	CPSW_NC_ETH_MAC_0_PN_MAC_TX_PAUSETIMER_REG	pn_mac_tx_pausetimer_reg	<a href="#">Go</a>
00022380h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_0	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
00022384h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_1	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
00022388h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_2	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>



**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0002238Ch	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_3	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
00022390h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_4	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
00022394h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_5	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
00022398h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_6	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
0002239Ch	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_PAUSETIMER_REG_7	pn_mac_txn_pausetimer_reg	<a href="#">Go</a>
000223A0h	CPSW_NC_ETH_MAC_0_PN_MAC EMC CONTROL_REG	pn_mac_emcontrol_reg	<a href="#">Go</a>
000223A4h	CPSW_NC_ETH_MAC_0_PN_MAC_TXN_GAP_REG	pn_mac_tx_gap_reg	<a href="#">Go</a>
000223A8h	CPSW_NC_ETH_MAC_0_PN_MAC_PORT_CONFIG	Port Configuration	<a href="#">Go</a>
000223ACh	CPSW_NC_ETH_MAC_0_PN_INTERVLAN_OPX_POINTER_REG	pn_opx_pointer_reg	<a href="#">Go</a>
000223B0h	CPSW_NC_ETH_MAC_0_PN_INTERVLAN_OPX_A_REG	pn_opx_a_reg	<a href="#">Go</a>
000223B4h	CPSW_NC_ETH_MAC_0_PN_INTERVLAN_OPX_B_REG	pn_opx_b_reg	<a href="#">Go</a>
000223B8h	CPSW_NC_ETH_MAC_0_PN_INTERVLAN_OPX_C_REG	pn_opx_c_reg	<a href="#">Go</a>
000223BCh	CPSW_NC_ETH_MAC_0_PN_INTERVLAN_OPX_D_REG	pn_opx_d_reg	<a href="#">Go</a>
00032000h	CPSW_NC_EST_FETCH_LOC_0	Revision Register	<a href="#">Go</a>
00032004h	CPSW_NC_EST_FETCH_LOC_1	Revision Register	<a href="#">Go</a>
00032008h	CPSW_NC_EST_FETCH_LOC_2	Revision Register	<a href="#">Go</a>
0003200Ch	CPSW_NC_EST_FETCH_LOC_3	Revision Register	<a href="#">Go</a>
00032010h	CPSW_NC_EST_FETCH_LOC_4	Revision Register	<a href="#">Go</a>
00032014h	CPSW_NC_EST_FETCH_LOC_5	Revision Register	<a href="#">Go</a>
00032018h	CPSW_NC_EST_FETCH_LOC_6	Revision Register	<a href="#">Go</a>
0003201Ch	CPSW_NC_EST_FETCH_LOC_7	Revision Register	<a href="#">Go</a>
00032020h	CPSW_NC_EST_FETCH_LOC_8	Revision Register	<a href="#">Go</a>
00032024h	CPSW_NC_EST_FETCH_LOC_9	Revision Register	<a href="#">Go</a>
00032028h	CPSW_NC_EST_FETCH_LOC_10	Revision Register	<a href="#">Go</a>
0003202Ch	CPSW_NC_EST_FETCH_LOC_11	Revision Register	<a href="#">Go</a>
00032030h	CPSW_NC_EST_FETCH_LOC_12	Revision Register	<a href="#">Go</a>
00032034h	CPSW_NC_EST_FETCH_LOC_13	Revision Register	<a href="#">Go</a>
00032038h	CPSW_NC_EST_FETCH_LOC_14	Revision Register	<a href="#">Go</a>
0003203Ch	CPSW_NC_EST_FETCH_LOC_15	Revision Register	<a href="#">Go</a>
00032040h	CPSW_NC_EST_FETCH_LOC_16	Revision Register	<a href="#">Go</a>
00032044h	CPSW_NC_EST_FETCH_LOC_17	Revision Register	<a href="#">Go</a>
00032048h	CPSW_NC_EST_FETCH_LOC_18	Revision Register	<a href="#">Go</a>
0003204Ch	CPSW_NC_EST_FETCH_LOC_19	Revision Register	<a href="#">Go</a>
00032050h	CPSW_NC_EST_FETCH_LOC_20	Revision Register	<a href="#">Go</a>
00032054h	CPSW_NC_EST_FETCH_LOC_21	Revision Register	<a href="#">Go</a>
00032058h	CPSW_NC_EST_FETCH_LOC_22	Revision Register	<a href="#">Go</a>



**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003205Ch	CPSW_NC_EST_FETCH_LOC_23	Revision Register	<a href="#">Go</a>
00032060h	CPSW_NC_EST_FETCH_LOC_24	Revision Register	<a href="#">Go</a>
00032064h	CPSW_NC_EST_FETCH_LOC_25	Revision Register	<a href="#">Go</a>
00032068h	CPSW_NC_EST_FETCH_LOC_26	Revision Register	<a href="#">Go</a>
0003206Ch	CPSW_NC_EST_FETCH_LOC_27	Revision Register	<a href="#">Go</a>
00032070h	CPSW_NC_EST_FETCH_LOC_28	Revision Register	<a href="#">Go</a>
00032074h	CPSW_NC_EST_FETCH_LOC_29	Revision Register	<a href="#">Go</a>
00032078h	CPSW_NC_EST_FETCH_LOC_30	Revision Register	<a href="#">Go</a>
0003207Ch	CPSW_NC_EST_FETCH_LOC_31	Revision Register	<a href="#">Go</a>
00032080h	CPSW_NC_EST_FETCH_LOC_32	Revision Register	<a href="#">Go</a>
00032084h	CPSW_NC_EST_FETCH_LOC_33	Revision Register	<a href="#">Go</a>
00032088h	CPSW_NC_EST_FETCH_LOC_34	Revision Register	<a href="#">Go</a>
0003208Ch	CPSW_NC_EST_FETCH_LOC_35	Revision Register	<a href="#">Go</a>
00032090h	CPSW_NC_EST_FETCH_LOC_36	Revision Register	<a href="#">Go</a>
00032094h	CPSW_NC_EST_FETCH_LOC_37	Revision Register	<a href="#">Go</a>
00032098h	CPSW_NC_EST_FETCH_LOC_38	Revision Register	<a href="#">Go</a>
0003209Ch	CPSW_NC_EST_FETCH_LOC_39	Revision Register	<a href="#">Go</a>
000320A0h	CPSW_NC_EST_FETCH_LOC_40	Revision Register	<a href="#">Go</a>
000320A4h	CPSW_NC_EST_FETCH_LOC_41	Revision Register	<a href="#">Go</a>
000320A8h	CPSW_NC_EST_FETCH_LOC_42	Revision Register	<a href="#">Go</a>
000320ACh	CPSW_NC_EST_FETCH_LOC_43	Revision Register	<a href="#">Go</a>
000320B0h	CPSW_NC_EST_FETCH_LOC_44	Revision Register	<a href="#">Go</a>
000320B4h	CPSW_NC_EST_FETCH_LOC_45	Revision Register	<a href="#">Go</a>
000320B8h	CPSW_NC_EST_FETCH_LOC_46	Revision Register	<a href="#">Go</a>
000320BCh	CPSW_NC_EST_FETCH_LOC_47	Revision Register	<a href="#">Go</a>
000320C0h	CPSW_NC_EST_FETCH_LOC_48	Revision Register	<a href="#">Go</a>
000320C4h	CPSW_NC_EST_FETCH_LOC_49	Revision Register	<a href="#">Go</a>
000320C8h	CPSW_NC_EST_FETCH_LOC_50	Revision Register	<a href="#">Go</a>
000320CCh	CPSW_NC_EST_FETCH_LOC_51	Revision Register	<a href="#">Go</a>
000320D0h	CPSW_NC_EST_FETCH_LOC_52	Revision Register	<a href="#">Go</a>
000320D4h	CPSW_NC_EST_FETCH_LOC_53	Revision Register	<a href="#">Go</a>
000320D8h	CPSW_NC_EST_FETCH_LOC_54	Revision Register	<a href="#">Go</a>
000320DCh	CPSW_NC_EST_FETCH_LOC_55	Revision Register	<a href="#">Go</a>
000320E0h	CPSW_NC_EST_FETCH_LOC_56	Revision Register	<a href="#">Go</a>
000320E4h	CPSW_NC_EST_FETCH_LOC_57	Revision Register	<a href="#">Go</a>
000320E8h	CPSW_NC_EST_FETCH_LOC_58	Revision Register	<a href="#">Go</a>
000320ECh	CPSW_NC_EST_FETCH_LOC_59	Revision Register	<a href="#">Go</a>
000320F0h	CPSW_NC_EST_FETCH_LOC_60	Revision Register	<a href="#">Go</a>
000320F4h	CPSW_NC_EST_FETCH_LOC_61	Revision Register	<a href="#">Go</a>
000320F8h	CPSW_NC_EST_FETCH_LOC_62	Revision Register	<a href="#">Go</a>
000320FCh	CPSW_NC_EST_FETCH_LOC_63	Revision Register	<a href="#">Go</a>
00032100h	CPSW_NC_EST_FETCH_LOC_64	Revision Register	<a href="#">Go</a>
00032104h	CPSW_NC_EST_FETCH_LOC_65	Revision Register	<a href="#">Go</a>
00032108h	CPSW_NC_EST_FETCH_LOC_66	Revision Register	<a href="#">Go</a>
0003210Ch	CPSW_NC_EST_FETCH_LOC_67	Revision Register	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00032110h	CPSW_NC_EST_FETCH_LOC_68	Revision Register	<a href="#">Go</a>
00032114h	CPSW_NC_EST_FETCH_LOC_69	Revision Register	<a href="#">Go</a>
00032118h	CPSW_NC_EST_FETCH_LOC_70	Revision Register	<a href="#">Go</a>
0003211Ch	CPSW_NC_EST_FETCH_LOC_71	Revision Register	<a href="#">Go</a>
00032120h	CPSW_NC_EST_FETCH_LOC_72	Revision Register	<a href="#">Go</a>
00032124h	CPSW_NC_EST_FETCH_LOC_73	Revision Register	<a href="#">Go</a>
00032128h	CPSW_NC_EST_FETCH_LOC_74	Revision Register	<a href="#">Go</a>
0003212Ch	CPSW_NC_EST_FETCH_LOC_75	Revision Register	<a href="#">Go</a>
00032130h	CPSW_NC_EST_FETCH_LOC_76	Revision Register	<a href="#">Go</a>
00032134h	CPSW_NC_EST_FETCH_LOC_77	Revision Register	<a href="#">Go</a>
00032138h	CPSW_NC_EST_FETCH_LOC_78	Revision Register	<a href="#">Go</a>
0003213Ch	CPSW_NC_EST_FETCH_LOC_79	Revision Register	<a href="#">Go</a>
00032140h	CPSW_NC_EST_FETCH_LOC_80	Revision Register	<a href="#">Go</a>
00032144h	CPSW_NC_EST_FETCH_LOC_81	Revision Register	<a href="#">Go</a>
00032148h	CPSW_NC_EST_FETCH_LOC_82	Revision Register	<a href="#">Go</a>
0003214Ch	CPSW_NC_EST_FETCH_LOC_83	Revision Register	<a href="#">Go</a>
00032150h	CPSW_NC_EST_FETCH_LOC_84	Revision Register	<a href="#">Go</a>
00032154h	CPSW_NC_EST_FETCH_LOC_85	Revision Register	<a href="#">Go</a>
00032158h	CPSW_NC_EST_FETCH_LOC_86	Revision Register	<a href="#">Go</a>
0003215Ch	CPSW_NC_EST_FETCH_LOC_87	Revision Register	<a href="#">Go</a>
00032160h	CPSW_NC_EST_FETCH_LOC_88	Revision Register	<a href="#">Go</a>
00032164h	CPSW_NC_EST_FETCH_LOC_89	Revision Register	<a href="#">Go</a>
00032168h	CPSW_NC_EST_FETCH_LOC_90	Revision Register	<a href="#">Go</a>
0003216Ch	CPSW_NC_EST_FETCH_LOC_91	Revision Register	<a href="#">Go</a>
00032170h	CPSW_NC_EST_FETCH_LOC_92	Revision Register	<a href="#">Go</a>
00032174h	CPSW_NC_EST_FETCH_LOC_93	Revision Register	<a href="#">Go</a>
00032178h	CPSW_NC_EST_FETCH_LOC_94	Revision Register	<a href="#">Go</a>
0003217Ch	CPSW_NC_EST_FETCH_LOC_95	Revision Register	<a href="#">Go</a>
00032180h	CPSW_NC_EST_FETCH_LOC_96	Revision Register	<a href="#">Go</a>
00032184h	CPSW_NC_EST_FETCH_LOC_97	Revision Register	<a href="#">Go</a>
00032188h	CPSW_NC_EST_FETCH_LOC_98	Revision Register	<a href="#">Go</a>
0003218Ch	CPSW_NC_EST_FETCH_LOC_99	Revision Register	<a href="#">Go</a>
00032190h	CPSW_NC_EST_FETCH_LOC_100	Revision Register	<a href="#">Go</a>
00032194h	CPSW_NC_EST_FETCH_LOC_101	Revision Register	<a href="#">Go</a>
00032198h	CPSW_NC_EST_FETCH_LOC_102	Revision Register	<a href="#">Go</a>
0003219Ch	CPSW_NC_EST_FETCH_LOC_103	Revision Register	<a href="#">Go</a>
000321A0h	CPSW_NC_EST_FETCH_LOC_104	Revision Register	<a href="#">Go</a>
000321A4h	CPSW_NC_EST_FETCH_LOC_105	Revision Register	<a href="#">Go</a>
000321A8h	CPSW_NC_EST_FETCH_LOC_106	Revision Register	<a href="#">Go</a>
000321ACh	CPSW_NC_EST_FETCH_LOC_107	Revision Register	<a href="#">Go</a>
000321B0h	CPSW_NC_EST_FETCH_LOC_108	Revision Register	<a href="#">Go</a>
000321B4h	CPSW_NC_EST_FETCH_LOC_109	Revision Register	<a href="#">Go</a>
000321B8h	CPSW_NC_EST_FETCH_LOC_110	Revision Register	<a href="#">Go</a>
000321BCh	CPSW_NC_EST_FETCH_LOC_111	Revision Register	<a href="#">Go</a>
000321C0h	CPSW_NC_EST_FETCH_LOC_112	Revision Register	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
000321C4h	CPSW_NC_EST_FETCH_LOC_113	Revision Register	<a href="#">Go</a>
000321C8h	CPSW_NC_EST_FETCH_LOC_114	Revision Register	<a href="#">Go</a>
000321CCh	CPSW_NC_EST_FETCH_LOC_115	Revision Register	<a href="#">Go</a>
000321D0h	CPSW_NC_EST_FETCH_LOC_116	Revision Register	<a href="#">Go</a>
000321D4h	CPSW_NC_EST_FETCH_LOC_117	Revision Register	<a href="#">Go</a>
000321D8h	CPSW_NC_EST_FETCH_LOC_118	Revision Register	<a href="#">Go</a>
000321DCh	CPSW_NC_EST_FETCH_LOC_119	Revision Register	<a href="#">Go</a>
000321E0h	CPSW_NC_EST_FETCH_LOC_120	Revision Register	<a href="#">Go</a>
000321E4h	CPSW_NC_EST_FETCH_LOC_121	Revision Register	<a href="#">Go</a>
000321E8h	CPSW_NC_EST_FETCH_LOC_122	Revision Register	<a href="#">Go</a>
000321ECh	CPSW_NC_EST_FETCH_LOC_123	Revision Register	<a href="#">Go</a>
000321F0h	CPSW_NC_EST_FETCH_LOC_124	Revision Register	<a href="#">Go</a>
000321F4h	CPSW_NC_EST_FETCH_LOC_125	Revision Register	<a href="#">Go</a>
000321F8h	CPSW_NC_EST_FETCH_LOC_126	Revision Register	<a href="#">Go</a>
000321FCh	CPSW_NC_EST_FETCH_LOC_127	Revision Register	<a href="#">Go</a>
00034000h	CPSW_CPDMA_REGS_CPDMA_FH_IDV ER_REG	CPDMA FHost IDVER	<a href="#">Go</a>
00034004h	CPSW_CPDMA_REGS_CPDMA_FH_CO NTROL_REG	CPDMA FHost Control Register	<a href="#">Go</a>
00034008h	CPSW_CPDMA_REGS_CPDMA_FH_TEA RDOWN_REG	CPDMA FHost Teardown Register	<a href="#">Go</a>
0003400Ch	CPSW_CPDMA_REGS_CPDMA_FH_EO Q_INT	CPDMA FHost Interrupt on EOQ only Register	<a href="#">Go</a>
00034010h	CPSW_CPDMA_REGS_CPDMA_TH_IDV ER_REG	CPDMA THost IDVER	<a href="#">Go</a>
00034014h	CPSW_CPDMA_REGS_CPDMA_TH_CO NTROL_REG	CPDMA THost Control Register	<a href="#">Go</a>
00034018h	CPSW_CPDMA_REGS_CPDMA_TH_TEA RDOWN_REG	CPDMA THost Teardown Register	<a href="#">Go</a>
0003401Ch	CPSW_CPDMA_REGS_CPDMA_SOFT_ RESET_REG	CPDMA Soft Reset Register	<a href="#">Go</a>
00034020h	CPSW_CPDMA_REGS_CPDMA_CONTR OL_REG	CPDMA Control Register	<a href="#">Go</a>
00034024h	CPSW_CPDMA_REGS_CPDMA_STATUS _REG	CPDMA Status Register	<a href="#">Go</a>
00034028h	CPSW_CPDMA_REGS_CPDMA_TH_BUF FER_OFFSET_REG	CPDMA THost Buffer Offset Register	<a href="#">Go</a>
0003402Ch	CPSW_CPDMA_REGS_CPDMA_EMULA TION_CONTROL_REG	CPDMA THost Buffer Offset Register	<a href="#">Go</a>
00034080h	CPSW_CPDMA_INT_CPDMA_FH_INTST AT_RAW_REG	CPDMA FHost Interrupt Status RAW	<a href="#">Go</a>
00034084h	CPSW_CPDMA_INT_CPDMA_FH_INTST AT_MASKED_REG	CPDMA FHost Interrupt Status MASKED	<a href="#">Go</a>
00034088h	CPSW_CPDMA_INT_CPDMA_FH_INTMA SK_SET_REG	CPDMA FHost Interrupt Masked Set	<a href="#">Go</a>
0003408Ch	CPSW_CPDMA_INT_CPDMA_FH_INTMA SK_CLEAR_REG	CPDMA FHost Interrupt Masked Clr	<a href="#">Go</a>
00034090h	CPSW_CPDMA_INT_CPDMA_IN_VECTO R_REG	CPDMA DMA IN Vector	<a href="#">Go</a>
00034094h	CPSW_CPDMA_INT_CPDMA_EOI_VECT OR_REG	CPDMA DMA EOI Vector	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
000340A0h	CPSW_CPDMA_INT_CPDMA_TH_INTST AT_RAW_REG	CPDMA receive Interrupt Status RAW	<a href="#">Go</a>
000340A4h	CPSW_CPDMA_INT_CPDMA_TH_INTST AT_MASKED_REG	CPDMA receive Interrupt Status MASKED	<a href="#">Go</a>
000340A8h	CPSW_CPDMA_INT_CPDMA_TH_INTMA SK_SET_REG	CPDMA receive Interrupt Status SET	<a href="#">Go</a>
000340ACh	CPSW_CPDMA_INT_CPDMA_TH_INTMA SK_CLEAR_REG	CPDMA receive Interrupt Status CLR	<a href="#">Go</a>
000340B0h	CPSW_CPDMA_INT_CPDMA_INTSTAT_ RAW_REG	CPDMA DMA Interrupt Status RAW	<a href="#">Go</a>
000340B4h	CPSW_CPDMA_INT_CPDMA_INTSTAT_ MASKED_REG	CPDMA DMA Interrupt Status MASKED	<a href="#">Go</a>
000340B8h	CPSW_CPDMA_INT_CPDMA_INTMASK_ SET_REG	CPDMA DMA Interrupt Status SET	<a href="#">Go</a>
000340BCh	CPSW_CPDMA_INT_CPDMA_INTMASK_ CLEAR_REG	CPDMA DMA Interrupt Status CLR	<a href="#">Go</a>
000340C0h	CPSW_CPDMA_INT_CPDMA_TH0_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340C4h	CPSW_CPDMA_INT_CPDMA_TH1_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340C8h	CPSW_CPDMA_INT_CPDMA_TH2_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340CCh	CPSW_CPDMA_INT_CPDMA_TH3_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340D0h	CPSW_CPDMA_INT_CPDMA_TH4_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340D4h	CPSW_CPDMA_INT_CPDMA_TH5_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340D8h	CPSW_CPDMA_INT_CPDMA_TH6_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340DCh	CPSW_CPDMA_INT_CPDMA_TH7_PEN DTHRESH_REG	CPDMA THost Threshold Pending Register	<a href="#">Go</a>
000340E0h	CPSW_CPDMA_INT_CPDMA_TH0_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340E4h	CPSW_CPDMA_INT_CPDMA_TH1_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340E8h	CPSW_CPDMA_INT_CPDMA_TH2_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340ECh	CPSW_CPDMA_INT_CPDMA_TH3_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340F0h	CPSW_CPDMA_INT_CPDMA_TH4_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340F4h	CPSW_CPDMA_INT_CPDMA_TH5_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340F8h	CPSW_CPDMA_INT_CPDMA_TH6_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
000340FCh	CPSW_CPDMA_INT_CPDMA_TH7_FREE BUFFER_REG	CPDMA THost Free Buffer Register	<a href="#">Go</a>
00034200h	CPSW_CPDMA_SRAM_CPDMA_FH0_H DP_REG	CPDMA FHost Channel 0 HDP	<a href="#">Go</a>
00034204h	CPSW_CPDMA_SRAM_CPDMA_FH1_H DP_REG	CPDMA FHost Channel 1 HDP	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00034208h	CPSW_CPDMA_SRAM_CPDMA_FH2_H DP_REG	CPDMA FHost Channel 2 HDP	<a href="#">Go</a>
0003420Ch	CPSW_CPDMA_SRAM_CPDMA_FH3_H DP_REG	CPDMA FHost Channel 3 HDP	<a href="#">Go</a>
00034210h	CPSW_CPDMA_SRAM_CPDMA_FH4_H DP_REG	CPDMA FHost Channel 4 HDP	<a href="#">Go</a>
00034214h	CPSW_CPDMA_SRAM_CPDMA_FH5_H DP_REG	CPDMA FHost Channel 5 HDP	<a href="#">Go</a>
00034218h	CPSW_CPDMA_SRAM_CPDMA_FH6_H DP_REG	CPDMA FHost Channel 6 HDP	<a href="#">Go</a>
0003421Ch	CPSW_CPDMA_SRAM_CPDMA_FH7_H DP_REG	CPDMA FHost Channel 7 HDP	<a href="#">Go</a>
00034220h	CPSW_CPDMA_SRAM_CPDMA_TH0_H DP_REG	CPDMA THost Channel 0 HDP	<a href="#">Go</a>
00034224h	CPSW_CPDMA_SRAM_CPDMA_TH1_H DP_REG	CPDMA THost Channel 1 HDP	<a href="#">Go</a>
00034228h	CPSW_CPDMA_SRAM_CPDMA_TH2_H DP_REG	CPDMA THost Channel 2 HDP	<a href="#">Go</a>
0003422Ch	CPSW_CPDMA_SRAM_CPDMA_TH3_H DP_REG	CPDMA THost Channel 3 HDP	<a href="#">Go</a>
00034230h	CPSW_CPDMA_SRAM_CPDMA_TH4_H DP_REG	CPDMA THost Channel 4 HDP	<a href="#">Go</a>
00034234h	CPSW_CPDMA_SRAM_CPDMA_TH5_H DP_REG	CPDMA THost Channel 5 HDP	<a href="#">Go</a>
00034238h	CPSW_CPDMA_SRAM_CPDMA_TH6_H DP_REG	CPDMA THost Channel 6 HDP	<a href="#">Go</a>
0003423Ch	CPSW_CPDMA_SRAM_CPDMA_TH7_H DP_REG	CPDMA THost Channel 7 HDP	<a href="#">Go</a>
00034240h	CPSW_CPDMA_SRAM_CPDMA_FH0_CP _REG	CPDMA FHost Channel 0 CP	<a href="#">Go</a>
00034244h	CPSW_CPDMA_SRAM_CPDMA_FH1_CP _REG	CPDMA FHost Channel 1 CP	<a href="#">Go</a>
00034248h	CPSW_CPDMA_SRAM_CPDMA_FH2_CP _REG	CPDMA FHost Channel 2 CP	<a href="#">Go</a>
0003424Ch	CPSW_CPDMA_SRAM_CPDMA_FH3_CP _REG	CPDMA FHost Channel 3 CP	<a href="#">Go</a>
00034250h	CPSW_CPDMA_SRAM_CPDMA_FH4_CP _REG	CPDMA FHost Channel 4 CP	<a href="#">Go</a>
00034254h	CPSW_CPDMA_SRAM_CPDMA_FH5_CP _REG	CPDMA FHost Channel 5 CP	<a href="#">Go</a>
00034258h	CPSW_CPDMA_SRAM_CPDMA_FH6_CP _REG	CPDMA FHost Channel 6 CP	<a href="#">Go</a>
0003425Ch	CPSW_CPDMA_SRAM_CPDMA_FH7_CP _REG	CPDMA FHost Channel 7 CP	<a href="#">Go</a>
00034260h	CPSW_CPDMA_SRAM_CPDMA_TH0_CP _REG	CPDMA THost Channel 0 CP	<a href="#">Go</a>
00034264h	CPSW_CPDMA_SRAM_CPDMA_TH1_CP _REG	CPDMA THost Channel 1 CP	<a href="#">Go</a>
00034268h	CPSW_CPDMA_SRAM_CPDMA_TH2_CP _REG	CPDMA THost Channel 2 CP	<a href="#">Go</a>
0003426Ch	CPSW_CPDMA_SRAM_CPDMA_TH3_CP _REG	CPDMA THost Channel 3 CP	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00034270h	CPSW_CPDMA_SRAM_CPDMA_TH4_CP_REG	CPDMA THost Channel 4 CP	<a href="#">Go</a>
00034274h	CPSW_CPDMA_SRAM_CPDMA_TH5_CP_REG	CPDMA THost Channel 5 CP	<a href="#">Go</a>
00034278h	CPSW_CPDMA_SRAM_CPDMA_TH6_CP_REG	CPDMA THost Channel 6 CP	<a href="#">Go</a>
0003427Ch	CPSW_CPDMA_SRAM_CPDMA_TH7_CP_REG	CPDMA THost Channel 7 CP	<a href="#">Go</a>
00034300h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H0_HDP_REG	Test CPDMA FHost Channel 0 HDP	<a href="#">Go</a>
00034304h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H1_HDP_REG	Test CPDMA FHost Channel 1 HDP	<a href="#">Go</a>
00034308h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H2_HDP_REG	Test CPDMA FHost Channel 2 HDP	<a href="#">Go</a>
0003430Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H3_HDP_REG	Test CPDMA FHost Channel 3 HDP	<a href="#">Go</a>
00034310h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H4_HDP_REG	Test CPDMA FHost Channel 4 HDP	<a href="#">Go</a>
00034314h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H5_HDP_REG	Test CPDMA FHost Channel 5 HDP	<a href="#">Go</a>
00034318h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H6_HDP_REG	Test CPDMA FHost Channel 6 HDP	<a href="#">Go</a>
0003431Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H7_HDP_REG	Test CPDMA FHost Channel 7 HDP	<a href="#">Go</a>
00034320h	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H0_HDP_REG	Test CPDMA THost Channel 0 HDP	<a href="#">Go</a>
00034324h	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H1_HDP_REG	Test CPDMA THost Channel 1 HDP	<a href="#">Go</a>
00034328h	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H2_HDP_REG	Test CPDMA THost Channel 2 HDP	<a href="#">Go</a>
0003432Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H3_HDP_REG	Test CPDMA THost Channel 3 HDP	<a href="#">Go</a>
00034330h	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H4_HDP_REG	Test CPDMA THost Channel 4 HDP	<a href="#">Go</a>
00034334h	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H5_HDP_REG	Test CPDMA THost Channel 5 HDP	<a href="#">Go</a>
00034338h	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H6_HDP_REG	Test CPDMA THost Channel 6 HDP	<a href="#">Go</a>
0003433Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_T_H7_HDP_REG	Test CPDMA THost Channel 7 HDP	<a href="#">Go</a>
00034340h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H0_CP_REG	Test CPDMA FHost Channel 0 CP	<a href="#">Go</a>
00034344h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H1_CP_REG	Test CPDMA FHost Channel 1 CP	<a href="#">Go</a>
00034348h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H2_CP_REG	Test CPDMA FHost Channel 2 CP	<a href="#">Go</a>
0003434Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H3_CP_REG	Test CPDMA FHost Channel 3 CP	<a href="#">Go</a>
00034350h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H4_CP_REG	Test CPDMA FHost Channel 4 CP	<a href="#">Go</a>
00034354h	CPSW_CPDMA_SRAM_TEST_CPDMA_F_H5_CP_REG	Test CPDMA FHost Channel 5 CP	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
00034358h	CPSW_CPDMA_SRAM_TEST_CPDMA_F H6_CP_REG	Test CPDMA FHost Channel 6 CP	<a href="#">Go</a>
0003435Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_F H7_CP_REG	Test CPDMA FHost Channel 7 CP	<a href="#">Go</a>
00034360h	CPSW_CPDMA_SRAM_TEST_CPDMA_T H0_CP_REG	Test CPDMA THost Channel 0 CP	<a href="#">Go</a>
00034364h	CPSW_CPDMA_SRAM_TEST_CPDMA_T H1_CP_REG	Test CPDMA THost Channel 1 CP	<a href="#">Go</a>
00034368h	CPSW_CPDMA_SRAM_TEST_CPDMA_T H2_CP_REG	Test CPDMA THost Channel 2 CP	<a href="#">Go</a>
0003436Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_T H3_CP_REG	Test CPDMA THost Channel 3 CP	<a href="#">Go</a>
00034370h	CPSW_CPDMA_SRAM_TEST_CPDMA_T H4_CP_REG	Test CPDMA THost Channel 4 CP	<a href="#">Go</a>
00034374h	CPSW_CPDMA_SRAM_TEST_CPDMA_T H5_CP_REG	Test CPDMA THost Channel 5 CP	<a href="#">Go</a>
00034378h	CPSW_CPDMA_SRAM_TEST_CPDMA_T H6_CP_REG	Test CPDMA THost Channel 6 CP	<a href="#">Go</a>
0003437Ch	CPSW_CPDMA_SRAM_TEST_CPDMA_T H7_CP_REG	Test CPDMA THost Channel 7 CP	<a href="#">Go</a>
0003A000h	CPSW_NC_STAT_0_RXGOODFRAMES	RxGoodFrames	<a href="#">Go</a>
0003A004h	CPSW_NC_STAT_0_RXBROADCASTFR AMES	RxBroadcastFrames	<a href="#">Go</a>
0003A008h	CPSW_NC_STAT_0_RXMULTICASTFRA MES	RxMulticastFrames	<a href="#">Go</a>
0003A010h	CPSW_NC_STAT_0_RXCRCERRORS	RxCRCErrors	<a href="#">Go</a>
0003A018h	CPSW_NC_STAT_0_RXOVERSIZEDFRA MES	RxOversizedFrames	<a href="#">Go</a>
0003A020h	CPSW_NC_STAT_0_RXUNDERSIZEDFR AMES	RxUndersizedFrames	<a href="#">Go</a>
0003A024h	CPSW_NC_STAT_0_RXFRAGMENTS	RxFragments	<a href="#">Go</a>
0003A028h	CPSW_NC_STAT_0_ALE_DROP	ALE_Drop	<a href="#">Go</a>
0003A02Ch	CPSW_NC_STAT_0_ALE_OVERRUN_DR OP	ALE_Overrun_Drop	<a href="#">Go</a>
0003A030h	CPSW_NC_STAT_0_RXOCTETS	RxOctets	<a href="#">Go</a>
0003A034h	CPSW_NC_STAT_0_TXGOODFRAMES	TxGoodFrames	<a href="#">Go</a>
0003A038h	CPSW_NC_STAT_0_TXBROADCASTFRA MES	TxBroadcastFrames	<a href="#">Go</a>
0003A03Ch	CPSW_NC_STAT_0_TXMULTICASTFRA MES	TxMulticastFrames	<a href="#">Go</a>
0003A04Ch	CPSW_NC_STAT_0_TXSINGLECOLLFR AMES	TxSingleCollFrames	<a href="#">Go</a>
0003A050h	CPSW_NC_STAT_0_TXMULTCOLLFRAM ES	TxMultCollFrames	<a href="#">Go</a>
0003A064h	CPSW_NC_STAT_0_TXOCTETS	TxOctets	<a href="#">Go</a>
0003A068h	CPSW_NC_STAT_0_OCTETFRAMES64	OctetFrames64	<a href="#">Go</a>
0003A06Ch	CPSW_NC_STAT_0_OCTETFRAMES65T 127	OctetFrames65t127	<a href="#">Go</a>
0003A070h	CPSW_NC_STAT_0_OCTETFRAMES128 T255	OctetFrames128t255	<a href="#">Go</a>
0003A074h	CPSW_NC_STAT_0_OCTETFRAMES256 T511	OctetFrames256t511	<a href="#">Go</a>



**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003A078h	CPSW_NC_STAT_0_OCTETFRAMES512 T1023	OctetFrames512t1023	<a href="#">Go</a>
0003A07Ch	CPSW_NC_STAT_0_OCTETFRAMES102 4TUP	OctetFrames1024tUP	<a href="#">Go</a>
0003A080h	CPSW_NC_STAT_0_NETOCTETS	NetOctets	<a href="#">Go</a>
0003A084h	CPSW_NC_STAT_0_RX_BOTTOM_OF_F IFO_DROP	Rx_Bottom_of_FIFO_Drop	<a href="#">Go</a>
0003A088h	CPSW_NC_STAT_0_PORTMASK_DROP	Portmask_Drop	<a href="#">Go</a>
0003A08Ch	CPSW_NC_STAT_0_RX_TOP_OF_FIFO_ DROP	Rx_Top_of_FIFO_Drop	<a href="#">Go</a>
0003A090h	CPSW_NC_STAT_0_ALE_RATE_LIMIT_D ROP	ALE_Rate_Limit_Drop	<a href="#">Go</a>
0003A094h	CPSW_NC_STAT_0_ALE_VID_INGRESS _DROP	ALE_VID_Ingress_Drop	<a href="#">Go</a>
0003A098h	CPSW_NC_STAT_0_ALE_DA_EQ_SA_D ROP	ALE_DA_EQ_SA_Drop	<a href="#">Go</a>
0003A09Ch	CPSW_NC_STAT_0_ALE_BLOCK_DROP	ALE_Block_Drop	<a href="#">Go</a>
0003A0A0h	CPSW_NC_STAT_0_ALE_SECURE_DRO P	ALE_Secure_Drop	<a href="#">Go</a>
0003A0A4h	CPSW_NC_STAT_0_ALE_AUTH_DROP	ALE_Auth_Drop	<a href="#">Go</a>
0003A0A8h	CPSW_NC_STAT_0_ALE_UNKN_UNI	ALE_Unkn_Uni	<a href="#">Go</a>
0003A0ACh	CPSW_NC_STAT_0_ALE_UNKN_UNI_BC NT	ALE_Unkn_Uni_Bcnt	<a href="#">Go</a>
0003A0B0h	CPSW_NC_STAT_0_ALE_UNKN_MLT	ALE_Unkn_Mlt	<a href="#">Go</a>
0003A0B4h	CPSW_NC_STAT_0_ALE_UNKN_MLT_B CNT	ALE_Unkn_Mlt_Bcnt	<a href="#">Go</a>
0003A0B8h	CPSW_NC_STAT_0_ALE_UNKN_BRD	ALE_Unkn_Brd	<a href="#">Go</a>
0003A0BCh	CPSW_NC_STAT_0_ALE_UNKN_BRD_B CNT	ALE_Unkn_Brd_Bcnt	<a href="#">Go</a>
0003A0C0h	CPSW_NC_STAT_0_ALE_POL_MATCH	ALE_Pol_Match	<a href="#">Go</a>
0003A0C4h	CPSW_NC_STAT_0_ALE_POL_MATCH_ RED	ALE_Pol_Match_Red	<a href="#">Go</a>
0003A0C8h	CPSW_NC_STAT_0_ALE_POL_MATCH_ YELLOW	ALE_Pol_Match_Yellow	<a href="#">Go</a>
0003A0CCh	CPSW_NC_STAT_0_ALE_MULT_SA_DR OP	ALE_MULT_SA_DROP	<a href="#">Go</a>
0003A0D0h	CPSW_NC_STAT_0_ALE_DUAL_VLAN_ DROP	ALE_DUAL_VLAN_DROP	<a href="#">Go</a>
0003A0D4h	CPSW_NC_STAT_0_ALE_LEN_ERROR_ DROP	ALE_LEN_ERROR_DROP	<a href="#">Go</a>
0003A0D8h	CPSW_NC_STAT_0_ALE_IP_NEXT_HDR _DROP	ALE_IP_NEXT_HDR_DROP	<a href="#">Go</a>
0003A0DCh	CPSW_NC_STAT_0_ALE_IPV4_FRAG_D ROP	ALE_IPV4_FRAG_DROP	<a href="#">Go</a>
0003A17Ch	CPSW_NC_STAT_0_TX_MEMORY_PRO TECT_ERROR	Tx_Memory_Protect_Error	<a href="#">Go</a>
0003A200h	CPSW_NC_STAT_1_RXGOODFRAMES	RxGoodFrames	<a href="#">Go</a>
0003A204h	CPSW_NC_STAT_1_RXBROADCASTFR AMES	RxBroadcastFrames	<a href="#">Go</a>
0003A208h	CPSW_NC_STAT_1_RXMULTICASTFRA MES	RxMulticastFrames	<a href="#">Go</a>
0003A20Ch	CPSW_NC_STAT_1_RXPAUSEFRAMES	RxPauseFrames	<a href="#">Go</a>



**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003A210h	CPSW_NC_STAT_1_RXCRCERRORS	RxCRCErrors	<a href="#">Go</a>
0003A214h	CPSW_NC_STAT_1_RXALIGNCODEERR ORS	RxAlignCodeErrors	<a href="#">Go</a>
0003A218h	CPSW_NC_STAT_1_RXOVERSIZEDFRA MES	RxOversizedFrames	<a href="#">Go</a>
0003A21Ch	CPSW_NC_STAT_1_RXJABBERFRAMES	RxJabberFrames	<a href="#">Go</a>
0003A220h	CPSW_NC_STAT_1_RXUNDERSIZEDFR AMES	RxUndersizedFrames	<a href="#">Go</a>
0003A224h	CPSW_NC_STAT_1_RXFRAGMENTS	RxFragments	<a href="#">Go</a>
0003A228h	CPSW_NC_STAT_1_ALE_DROP	ALE_Drop	<a href="#">Go</a>
0003A22Ch	CPSW_NC_STAT_1_ALE_OVERRUN_DR OP	ALE_Overrun_Drop	<a href="#">Go</a>
0003A230h	CPSW_NC_STAT_1_RXOCTETS	RxOctets	<a href="#">Go</a>
0003A234h	CPSW_NC_STAT_1_TXGOODFRAMES	TxGoodFrames	<a href="#">Go</a>
0003A238h	CPSW_NC_STAT_1_TXBROADCASTFRA MES	TxBroadcastFrames	<a href="#">Go</a>
0003A23Ch	CPSW_NC_STAT_1_TXMULTICASTFRA MES	TxMulticastFrames	<a href="#">Go</a>
0003A240h	CPSW_NC_STAT_1_TXPAUSEFRAMES	TxPauseFrames	<a href="#">Go</a>
0003A244h	CPSW_NC_STAT_1_TXDEFERREDFRA MES	TxDeferredFrames	<a href="#">Go</a>
0003A248h	CPSW_NC_STAT_1_TXCOLLISIONFRAM ES	TxCollisionFrames	<a href="#">Go</a>
0003A24Ch	CPSW_NC_STAT_1_TXSINGLECOLLFR AMES	TxSingleCollFrames	<a href="#">Go</a>
0003A250h	CPSW_NC_STAT_1_TXMULTCOLLFRAM ES	TxMultCollFrames	<a href="#">Go</a>
0003A254h	CPSW_NC_STAT_1_TXEXCESSIVECOL LISIONS	TxExcessiveCollisions	<a href="#">Go</a>
0003A258h	CPSW_NC_STAT_1_TXLATECOLLISION S	TxLateCollisions	<a href="#">Go</a>
0003A25Ch	CPSW_NC_STAT_1_RXIPGERROR	RxIPGError	<a href="#">Go</a>
0003A260h	CPSW_NC_STAT_1_TXCARRIERSENSE ERRORS	TxCARRIERSENSEErrors	<a href="#">Go</a>
0003A264h	CPSW_NC_STAT_1_TXOCTETS	TxOctets	<a href="#">Go</a>
0003A268h	CPSW_NC_STAT_1_OCTETFRAMES64	OctetFrames64	<a href="#">Go</a>
0003A26Ch	CPSW_NC_STAT_1_OCTETFRAMES65T 127	OctetFrames65t127	<a href="#">Go</a>
0003A270h	CPSW_NC_STAT_1_OCTETFRAMES128 T255	OctetFrames128t255	<a href="#">Go</a>
0003A274h	CPSW_NC_STAT_1_OCTETFRAMES256 T511	OctetFrames256t511	<a href="#">Go</a>
0003A278h	CPSW_NC_STAT_1_OCTETFRAMES512 T1023	OctetFrames512t1023	<a href="#">Go</a>
0003A27Ch	CPSW_NC_STAT_1_OCTETFRAMES102 4TUP	OctetFrames1024tUP	<a href="#">Go</a>
0003A280h	CPSW_NC_STAT_1_NETOCTETS	NetOctets	<a href="#">Go</a>
0003A284h	CPSW_NC_STAT_1_RX_BOTTOM_OF_F IFO_DROP	Rx_Bottom_of_FIFO_Drop	<a href="#">Go</a>
0003A288h	CPSW_NC_STAT_1_PORTMASK_DROP	Portmask_Drop	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003A28Ch	CPSW_NC_STAT_1_RX_TOP_OF_FIFO_DROP	Rx_Top_of_FIFO_Drop	<a href="#">Go</a>
0003A290h	CPSW_NC_STAT_1_ALE_RATE_LIMIT_DROP	ALE_Rate_Limit_Drop	<a href="#">Go</a>
0003A294h	CPSW_NC_STAT_1_ALE_VID_INGRESS_DROP	ALE_VID_Ingress_Drop	<a href="#">Go</a>
0003A298h	CPSW_NC_STAT_1_ALE_DA_EQ_SA_DROP	ALE_DA_EQ_SA_Drop	<a href="#">Go</a>
0003A29Ch	CPSW_NC_STAT_1_ALE_BLOCK_DROP	ALE_Block_Drop	<a href="#">Go</a>
0003A2A0h	CPSW_NC_STAT_1_ALE_SECURE_DROP	ALE_Secure_Drop	<a href="#">Go</a>
0003A2A4h	CPSW_NC_STAT_1_ALE_AUTH_DROP	ALE_Auth_Drop	<a href="#">Go</a>
0003A2A8h	CPSW_NC_STAT_1_ALE_UNKN_UNI	ALE_Unkn_Uni	<a href="#">Go</a>
0003A2ACh	CPSW_NC_STAT_1_ALE_UNKN_UNI_BCNT	ALE_Unkn_Uni_Bcnt	<a href="#">Go</a>
0003A2B0h	CPSW_NC_STAT_1_ALE_UNKN_MLT	ALE_Unkn_Mlt	<a href="#">Go</a>
0003A2B4h	CPSW_NC_STAT_1_ALE_UNKN_MLT_BCNT	ALE_Unkn_Mlt_Bcnt	<a href="#">Go</a>
0003A2B8h	CPSW_NC_STAT_1_ALE_UNKN_BRD	ALE_Unkn_Brd	<a href="#">Go</a>
0003A2BCh	CPSW_NC_STAT_1_ALE_UNKN_BRD_BCNT	ALE_Unkn_Brd_Bcnt	<a href="#">Go</a>
0003A2C0h	CPSW_NC_STAT_1_ALE_POL_MATCH	ALE_Pol_Match	<a href="#">Go</a>
0003A2C4h	CPSW_NC_STAT_1_ALE_POL_MATCH_RED	ALE_Pol_Match_Red	<a href="#">Go</a>
0003A2C8h	CPSW_NC_STAT_1_ALE_POL_MATCH_YELLOW	ALE_Pol_Match_Yellow	<a href="#">Go</a>
0003A2CCh	CPSW_NC_STAT_1_ALE_MULT_SA_DROP	ALE_MULT_SA_DROP	<a href="#">Go</a>
0003A2D0h	CPSW_NC_STAT_1_ALE_DUAL_VLAN_DROP	ALE_DUAL_VLAN_DROP	<a href="#">Go</a>
0003A2D4h	CPSW_NC_STAT_1_ALE_LEN_ERROR_DROP	ALE_LEN_ERROR_DROP	<a href="#">Go</a>
0003A2D8h	CPSW_NC_STAT_1_ALE_IP_NEXT_HDR_DROP	ALE_IP_NEXT_HDR_DROP	<a href="#">Go</a>
0003A2DCh	CPSW_NC_STAT_1_ALE_IPV4_FRAG_DROP	ALE_IPV4_FRAG_DROP	<a href="#">Go</a>
0003A37Ch	CPSW_NC_STAT_1_TX_MEMORY_PROTECT_ERROR	Tx_Memory_Protect_Error	<a href="#">Go</a>
0003A380h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_0	enet_pn_tx_pri	<a href="#">Go</a>
0003A384h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_1	enet_pn_tx_pri	<a href="#">Go</a>
0003A388h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_2	enet_pn_tx_pri	<a href="#">Go</a>
0003A38Ch	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_3	enet_pn_tx_pri	<a href="#">Go</a>
0003A390h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_4	enet_pn_tx_pri	<a href="#">Go</a>
0003A394h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_5	enet_pn_tx_pri	<a href="#">Go</a>
0003A398h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_REG_6	enet_pn_tx_pri	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003A39Ch	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ REG_7	enet_pn_tx_pri	<a href="#">Go</a>
0003A3A0h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_0	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3A4h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_1	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3A8h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_2	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3ACh	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_3	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3B0h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_4	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3B4h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_5	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3B8h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_6	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3BCh	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ BCNT_REG_7	enet_pn_tx_pri_bcmt	<a href="#">Go</a>
0003A3C0h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_0	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3C4h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_1	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3C8h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_2	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3CCh	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_3	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3D0h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_4	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3D4h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_5	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3D8h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_6	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3DCh	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_REG_7	enet_pn_tx_pri_drop	<a href="#">Go</a>
0003A3E0h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_0	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3E4h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_1	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3E8h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_2	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3ECh	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_3	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3F0h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_4	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3F4h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_5	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3F8h	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_6	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003A3FCh	CPSW_NC_STAT_1_ENET_PN_TX_PRI_ DROP_BCNT_REG_7	enet_pn_tx_pri_drop_bcmt	<a href="#">Go</a>
0003D000h	IDVER_REG	idver_reg	<a href="#">Go</a>
0003D004h	CPTS_CONTROL_REG	control_reg	<a href="#">Go</a>
0003D008h	CPTS_RFTCLK_SEL_REG	rftclk_sel_reg	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003D00Ch	CPTS_TS_PUSH_REG	ts_push_reg	<a href="#">Go</a>
0003D010h	TS_LOAD_VAL_REG	ts_load_low_val_reg	<a href="#">Go</a>
0003D014h	CPTS_TS_LOAD_EN_REG	ts_load_en_reg	<a href="#">Go</a>
0003D018h	TS_COMP_VAL_REG	ts_comp_low_val_reg	<a href="#">Go</a>
0003D01Ch	CPTS_TS_COMP_LEN_REG	ts_comp_len_reg	<a href="#">Go</a>
0003D020h	CPTS_INTSTAT_RAW_REG	intstat_raw_reg	<a href="#">Go</a>
0003D024h	CPTS_INTSTAT_MASKED_REG	intstat_masked_reg	<a href="#">Go</a>
0003D028h	CPTS_INT_ENABLE_REG	int_enable_reg	<a href="#">Go</a>
0003D02Ch	CPTS_TS_COMP_NUDGE_REG	ts_comp_nudge_reg	<a href="#">Go</a>
0003D030h	CPTS_EVENT_POP_REG	event_pop_reg	<a href="#">Go</a>
0003D034h	CPTS_EVENT_0_REG	event_0_reg	<a href="#">Go</a>
0003D038h	CPTS_EVENT_1_REG	event_1_reg	<a href="#">Go</a>
0003D03Ch	CPTS_EVENT_2_REG	event_2_reg	<a href="#">Go</a>
0003D040h	CPTS_EVENT_3_REG	event_3_reg	<a href="#">Go</a>
0003D044h	CPTS_TS_LOAD_HIGH_VAL_REG	ts_load_high_val_reg	<a href="#">Go</a>
0003D048h	CPTS_TS_COMP_HIGH_VAL_REG	ts_comp_high_val_reg	<a href="#">Go</a>
0003D04Ch	CPTS_TS_ADD_VAL_REG	ts_add_val	<a href="#">Go</a>
0003D050h	CPTS_TS_PPM_LOW_VAL_REG	ts_ppm_low_val_reg	<a href="#">Go</a>
0003D054h	CPTS_TS_PPM_HIGH_VAL_REG	ts_ppm_high_val_reg	<a href="#">Go</a>
0003D058h	CPTS_TS_NUDGE_VAL_REG	ts_nudge_val_reg	<a href="#">Go</a>
0003D0D0h	CPTS_TS_CONFIG	ts_config	<a href="#">Go</a>
0003D0E0h	TS_GENF0_COMP_LOW_REG	comp_low_reg	<a href="#">Go</a>
0003D0E4h	TS_GENF0_COMP_HIGH_REG	comp_high_reg	<a href="#">Go</a>
0003D0E8h	TS_GENF0_CONTROL_REG	control_reg	<a href="#">Go</a>
0003D0ECh	TS_GENF0_LENGTH_REG	length_reg	<a href="#">Go</a>
0003D0F0h	TS_GENF0_PPM_LOW_REG	ppm_low_reg	<a href="#">Go</a>
0003D0F4h	TS_GENF0_PPM_HIGH_REG	ppm_high_reg	<a href="#">Go</a>
0003D0F8h	TS_GENF0_NUDGE_REG	nudge_reg	<a href="#">Go</a>
0003D100h	TS_GENF1_COMP_LOW_REG	comp_low_reg	<a href="#">Go</a>
0003D104h	TS_GENF1_COMP_HIGH_REG	comp_high_reg	<a href="#">Go</a>
0003D108h	TS_GENF1_CONTROL_REG	control_reg	<a href="#">Go</a>
0003D10Ch	TS_GENF1_LENGTH_REG	length_reg	<a href="#">Go</a>
0003D110h	TS_GENF1_PPM_LOW_REG	ppm_low_reg	<a href="#">Go</a>
0003D114h	TS_GENF1_PPM_HIGH_REG	ppm_high_reg	<a href="#">Go</a>
0003D118h	TS_GENF1_NUDGE_REG	nudge_reg	<a href="#">Go</a>
0003D120h	TS_GENF2_COMP_LOW_REG	comp_low_reg	<a href="#">Go</a>
0003D124h	TS_GENF2_COMP_HIGH_REG	comp_high_reg	<a href="#">Go</a>
0003D128h	TS_GENF2_CONTROL_REG	control_reg	<a href="#">Go</a>
0003D12Ch	TS_GENF2_LENGTH_REG	length_reg	<a href="#">Go</a>
0003D130h	TS_GENF2_PPM_LOW_REG	ppm_low_reg	<a href="#">Go</a>
0003D134h	TS_GENF2_PPM_HIGH_REG	ppm_high_reg	<a href="#">Go</a>
0003D138h	TS_GENF2_NUDGE_REG	nudge_reg	<a href="#">Go</a>
0003D200h	TS_ESTF_COMP_LOW_REG	comp_low_reg	<a href="#">Go</a>
0003D204h	TS_ESTF_COMP_HIGH_REG	comp_high_reg	<a href="#">Go</a>
0003D208h	TS_ESTF_CONTROL_REG	control_reg	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003D20Ch	TS_ESTF_LENGTH_REG	length_reg	<a href="#">Go</a>
0003D210h	TS_ESTF_PPM_LOW_REG	ppm_low_reg	<a href="#">Go</a>
0003D214h	TS_ESTF_PPM_HIGH_REG	ppm_high_reg	<a href="#">Go</a>
0003D218h	TS_ESTF_NUDGE_REG	nudge_reg	<a href="#">Go</a>
0003E000h	ALE_MOD_VER	Module and Version	<a href="#">Go</a>
0003E004h	ALE_ALE_STATUS	ALE Status	<a href="#">Go</a>
0003E008h	ALE_ALE_CONTROL	ALE Control	<a href="#">Go</a>
0003E00Ch	ALE_ALE_CTRL2	ALE Control 2	<a href="#">Go</a>
0003E010h	ALE_ALE_PRESCALE	ALE Prescale	<a href="#">Go</a>
0003E014h	ALE_ALE_AGING_CTRL	ALE Aging Control	<a href="#">Go</a>
0003E01Ch	ALE_ALE_NXT_HDR	ALE Next Header	<a href="#">Go</a>
0003E020h	ALE_ALE_TBLCTL	ALE Table Control	<a href="#">Go</a>
0003E034h	ALE_ALE_TBLW2	ALE LUT Table word 2	<a href="#">Go</a>
0003E038h	ALE_ALE_TBLW1	ALE LUT Table word 1	<a href="#">Go</a>
0003E03Ch	ALE_ALE_TBLW0	ALE LUT Table word 0	<a href="#">Go</a>
0003E040h	ALE_I0_ALE_PORTCTL0_0	ALE Port Control X	<a href="#">Go</a>
0003E044h	ALE_I0_ALE_PORTCTL0_1	ALE Port Control X	<a href="#">Go</a>
0003E090h	ALE_ALE_UVLAN_MEMBER	ALE Unknown VLAN Member Mask Register	<a href="#">Go</a>
0003E094h	ALE_ALE_UVLAN_URCAST	ALE Unknown VLAN Unregistered Multicast Flood Mask Register	<a href="#">Go</a>
0003E098h	ALE_ALE_UVLAN_RMCAST	ALE Unknown VLAN Registered Multicast Flood Mask Register	<a href="#">Go</a>
0003E09Ch	ALE_ALE_UVLAN_UNTAG	ALE Unknown VLAN force Untagged Egress Mask Register	<a href="#">Go</a>
0003E0B4h	ALE_ALE_FAST_LUT	ALE Fast LUT Register	<a href="#">Go</a>
0003E0B8h	ALE_ALE_STAT_DIAG	ALE Statistic Output Diagnostic Register	<a href="#">Go</a>
0003E0BCh	ALE_ALE_OAM_LB_CTRL	ALE OAM Loopback Control	<a href="#">Go</a>
0003E0C0h	ALE_ALE_MSK_MUX0	ALE Mask Mux 0	<a href="#">Go</a>
0003E0C4h	ALE_I1_ALE_MSK_MUX1_0	ALE Mask Mux X	<a href="#">Go</a>
0003E0C8h	ALE_I1_ALE_MSK_MUX1_1	ALE Mask Mux X	<a href="#">Go</a>
0003E0CCh	ALE_I1_ALE_MSK_MUX1_2	ALE Mask Mux X	<a href="#">Go</a>
0003E0FCh	ALE_EGRESSOP	Egress Operation	<a href="#">Go</a>
0003E100h	ALE_POLICECFG0	Policing Config 0	<a href="#">Go</a>
0003E104h	ALE_POLICECFG1	Policing Config 1	<a href="#">Go</a>
0003E108h	ALE_POLICECFG2	Policing Config 2	<a href="#">Go</a>
0003E10Ch	ALE_POLICECFG3	Policing Config 3	<a href="#">Go</a>
0003E110h	ALE_POLICECFG4	Policing Config 4	<a href="#">Go</a>
0003E118h	ALE_POLICECFG6	Policing Config 6	<a href="#">Go</a>
0003E11Ch	ALE_POLICECFG7	Policing Config 7	<a href="#">Go</a>
0003E120h	ALE_POLICETBLCTL	Policing Table Control	<a href="#">Go</a>
0003E124h	ALE_POLICECONTROL	Policing Control	<a href="#">Go</a>
0003E128h	ALE_POLICETESTCTL	Policing Test Control	<a href="#">Go</a>
0003E12Ch	ALE_POLICEHSTAT	Policing Hit Status	<a href="#">Go</a>
0003E134h	ALE_THREADMAPDEF	THREAD Mapping Default Value	<a href="#">Go</a>
0003E138h	ALE_THREADMAPCTL	THREAD Mapping Control	<a href="#">Go</a>
0003E13Ch	ALE_THREADMAPVAL	THREAD Mapping Value	<a href="#">Go</a>

**Table 9-31. MSS\_CPSW Registers (continued)**

Offset	Acronym	Register Name	Section
0003F000h	rev	Aggregator Revision Register	<a href="#">Go</a>
0003F008h	vector	ECC Vector Register	<a href="#">Go</a>
0003F00Ch	stat	Misc Status	<a href="#">Go</a>
0003F010h	ECC_reserved_svbus_0	Reserved Area for Serial VBUS Registers	
0003F014h	ECC_reserved_svbus_1	Reserved Area for Serial VBUS Registers	
0003F018h	ECC_reserved_svbus_2	Reserved Area for Serial VBUS Registers	
0003F01Ch	ECC_reserved_svbus_3	Reserved Area for Serial VBUS Registers	
0003F020h	ECC_reserved_svbus_4	Reserved Area for Serial VBUS Registers	
0003F024h	ECC_reserved_svbus_5	Reserved Area for Serial VBUS Registers	
0003F028h	ECC_reserved_svbus_6	Reserved Area for Serial VBUS Registers	
0003F02Ch	ECC_reserved_svbus_7	Reserved Area for Serial VBUS Registers	
0003F03Ch	ECC_sec_eoi_reg	EOI Register	<a href="#">Go</a>
0003F040h	ECC_sec_status_reg0	Interrupt Status Register 0	<a href="#">Go</a>
0003F080h	ECC_sec_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Go</a>
0003F0C0h	ECC_sec_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Go</a>
0003F13Ch	ECC_ded_eoi_reg	EOI Register	<a href="#">Go</a>
0003F140h	ECC_ded_status_reg0	Interrupt Status Register 0	<a href="#">Go</a>
0003F180h	ECC_ded_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Go</a>
0003F1C0h	ECC_ded_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Go</a>
0003F200h	aggr_enable_set	AGGR interrupt enable set Register	<a href="#">Go</a>
0003F204h	aggr_enable_clr	AGGR interrupt enable clear Register	<a href="#">Go</a>
0003F208h	aggr_status_set	AGGR interrupt status set Register	<a href="#">Go</a>
0003F20Ch	aggr_status_clr	AGGR interrupt status clear Register	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 9-32](#) shows the codes that are used for access types in this section.

**Table 9-32. MSS\_CPSW Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Wdecr	W decr	Write
Wincr	W incr	Write
Reset or Default Value		
-n		Value after reset or the default value

### 9.6.1 CPSW\_NUSS\_IDVER\_REG Register (Offset = 0h) [Reset = 6BA01903h]

CPSW\_NUSS\_IDVER\_REG is shown in [Table 9-33](#).

Return to the [Summary Table](#).

ID Version Register

**Table 9-33. CPSW\_NUSS\_IDVER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	IDENT	R	6BA0h	Identification value
15-11	RTL_VER	R	3h	RTL version value
10-8	MAJOR_VER	R	1h	Major version value
7-0	MINOR_VER	R	3h	Minor version value

### 9.6.2 SS\_SYNCE\_COUNT\_REG Register (Offset = 4h) [Reset = 00000000h]

SS\_SYNCE\_COUNT\_REG is shown in [Table 9-34](#).

Return to the [Summary Table](#).

SS SYNCE Count Register

**Table 9-34. SS\_SYNCE\_COUNT\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SYNCE_CNT	R/W	0h	Sync E Count Value



### 9.6.3 SS\_SYNCE\_MUX\_REG Register (Offset = 8h) [Reset = 00000000h]

SS\_SYNCE\_MUX\_REG is shown in [Table 9-35](#).

Return to the [Summary Table](#).

SS Synce Mux Register

**Table 9-35. SS\_SYNCE\_MUX\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	SYNCE_SEL	R/W	0h	Sync E Select Value

### 9.6.4 SS\_CONTROL\_REG Register (Offset = Ch) [Reset = 00000000h]

SS\_CONTROL\_REG is shown in [Table 9-36](#).

Return to the [Summary Table](#).

SS Control Register

**Table 9-36. SS\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	EEE_PHY_ONLY	R/W	0h	Energy Efficient Enable Phy Only Mode: 0=The low power indicate state includes gating off the CPPI_GCLK to the CPSW, 1=The low power indicate state does not gate the clock to the CPSW
0	EEE_EN	R/W	0h	Energy Efficient Ethernet Enable: 0=EEE is disabled, 1=EEE is enabled

### 9.6.5 SS\_INT\_CONTROL\_REG Register (Offset = 18h) [Reset = XXX0X000h]

SS\_INT\_CONTROL\_REG is shown in [Table 9-37](#).

Return to the [Summary Table](#).

SS Interrupt Control Register

**Table 9-37. SS\_INT\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	INT_TEST	R/W	0h	Interrupt Test
30	INT_SEL_VEC_EN	R/W	0h	Interrupt Sel Vector Enable
29-22	RESERVED	R	0h	
21-16	INT_BYPASS	R/W	0h	Interrupt Bypass Value
15-12	RESERVED	R	0h	
11-0	INT_PRESCALE	R/W	0h	Interrupt Prescale Value

### 9.6.6 SS\_STATUS\_REG Register (Offset = 1Ch) [Reset = 00000000h]

SS\_STATUS\_REG is shown in [Table 9-38](#).

Return to the [Summary Table](#).

SS Status Register

**Table 9-38. SS\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	EEE_CLKSTOP_ACK	R	0h	Energy Efficient Ethernet clockstop acknowledge from CPSW

### 9.6.7 SUBSYSTEM\_CONFIG\_REG Register (Offset = 20h) [Reset = 0003X302h]

SUBSYSTEM\_CONFIG\_REG is shown in [Table 9-39](#).

Return to the [Summary Table](#).

Subsystem Configuration Register

**Table 9-39. SUBSYSTEM\_CONFIG\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-20	XGMII	R	0h	The Number of XGMII Ports included in the CPSW_NUSS
19	QSGMII	R	0h	QSGMII is included in the CPSW_NUSS
18	SGMII	R	0h	SGMII is included in the CPSW_NUSS
17	RGMII	R	1h	RGMII is included in the CPSW_NUSS
16	RMII	R	1h	RMII is included in the CPSW_NUSS
15-13	RESERVED	R	0h	
12-8	NUM_GENF	R	3h	The number of CPTS GENF outputs
7-0	NUM_PORTS	R	2h	The total number of ports including the host port 0

### 9.6.8 RGMII1\_STATUS\_REG Register (Offset = 30h) [Reset = 00000000h]

RGMII1\_STATUS\_REG is shown in [Table 9-40](#).

Return to the [Summary Table](#).

RGMII1 Status Register

**Table 9-40. RGMII1\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3	FULLDUPLEX	R	0h	Rgmii 1 full dulex: 0=Half-duplex, 1=Full-duplex
2-1	SPEED	R	0h	Rgmii 1 speed: 00= 10Mbps, 01= 100Mbps, 10= 1000Mbps, 11=reserved
0	LINK	R	0h	Rgmii 1 link indicator: 0=Link is down, 1=Link is up

### 9.6.9 MDIO\_MDIO\_VERSION\_REG Register (Offset = F00h) [Reset = 00071107h]

MDIO\_MDIO\_VERSION\_REG is shown in [Table 9-41](#).

Return to the [Summary Table](#).

MDIO Version Register

**Table 9-41. MDIO\_MDIO\_VERSION\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	0h	Scheme
29-28	bu	R	0h	bu
27-16	module_id	R	7h	Module ID
15-11	revrtl	R	2h	RTL version
10-8	revmaj	R	1h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	7h	Minor version

### 9.6.10 MDIO\_CONTROL\_REG Register (Offset = F04h) [Reset = X1XX00FFh]

MDIO\_CONTROL\_REG is shown in [Table 9-42](#).

Return to the [Summary Table](#).

MDIO Control Register

**Table 9-42. MDIO\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	MDIO state machine idle
30	ENABLE	R/W	0h	Enable control
29	RESERVED	R	0h	
28-24	HIGHEST_USER_CHANNEL	R	1h	Highest user channel
23-21	RESERVED	R	0h	
20	PREAMBLE	R/W	0h	Preamble disable
19	FAULT	R/W	0h	Fault indicator
18	FAULT_DETECT_ENABLE	R/W	0h	Fault detect enable
17	INT_TEST_ENABLE	R/W	0h	Interrupt test enable
16	RESERVED	R	0h	
15-0	CLKDIV	R/W	FFh	Clock divider



### 9.6.11 MDIO\_ALIVE\_REG Register (Offset = F08h) [Reset = 00000000h]

MDIO\_ALIVE\_REG is shown in [Table 9-43](#).

Return to the [Summary Table](#).

MDIO Alive Register

**Table 9-43. MDIO\_ALIVE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ALIVE	R/W	0h	MDIO alive

### 9.6.12 MDIO\_LINK\_REG Register (Offset = F0Ch) [Reset = 00000000h]

MDIO\_LINK\_REG is shown in [Table 9-44](#).

Return to the [Summary Table](#).

MDIO Link Register

**Table 9-44. MDIO\_LINK\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LINK	R	0h	MDIO link state

### 9.6.13 MDIO\_LINK\_INT\_RAW\_REG Register (Offset = F10h) [Reset = 00000000h]

MDIO\_LINK\_INT\_RAW\_REG is shown in [Table 9-45](#).

Return to the [Summary Table](#).

MDIO Link Interrupt Raw Register

**Table 9-45. MDIO\_LINK\_INT\_RAW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	LINKINTRAW	R/W	0h	MDIO link change event raw value

### 9.6.14 MDIO\_LINK\_INT\_MASKED\_REG Register (Offset = F14h) [Reset = 00000000h]

MDIO\_LINK\_INT\_MASKED\_REG is shown in [Table 9-46](#).

Return to the [Summary Table](#).

MDIO Link Interrupt Masked Register

**Table 9-46. MDIO\_LINK\_INT\_MASKED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	LINKINTMASKED	R/W	0h	MDIO link change interrupt masked value

### 9.6.15 MDIO\_LINK\_INT\_MASK\_SET\_REG Register (Offset = F18h) [Reset = 00000000h]

MDIO\_LINK\_INT\_MASK\_SET\_REG is shown in [Table 9-47](#).

Return to the [Summary Table](#).

MDIO Link Interrupt Mask Set Register

**Table 9-47. MDIO\_LINK\_INT\_MASK\_SET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	LINKINTMASKSET	R/W	0h	MDIO link interrupt mask set

### 9.6.16 MDIO\_LINK\_INT\_MASK\_CLEAR\_REG Register (Offset = F1Ch) [Reset = 00000000h]

MDIO\_LINK\_INT\_MASK\_CLEAR\_REG is shown in [Table 9-48](#).

Return to the [Summary Table](#).

MDIO Link Interrupt Mask Clear Register

**Table 9-48. MDIO\_LINK\_INT\_MASK\_CLEAR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	LINKINTMASKCLR	R/W	0h	MDIO link interrupt mask clear

### 9.6.17 MDIO\_USER\_INT\_RAW\_REG Register (Offset = F20h) [Reset = 00000000h]

MDIO\_USER\_INT\_RAW\_REG is shown in [Table 9-49](#).

Return to the [Summary Table](#).

MDIO User Interrupt Raw Register

**Table 9-49. MDIO\_USER\_INT\_RAW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	USERINTRAW	R/W	0h	User interrupt raw

### 9.6.18 MDIO\_USER\_INT\_MASKED\_REG Register (Offset = F24h) [Reset = 00000000h]

MDIO\_USER\_INT\_MASKED\_REG is shown in [Table 9-50](#).

Return to the [Summary Table](#).

MDIO User Interrupt Masked Register

**Table 9-50. MDIO\_USER\_INT\_MASKED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	USERINTMASKED	R/W	0h	User interrupt masked



### 9.6.19 MDIO\_USER\_INT\_MASK\_SET\_REG Register (Offset = F28h) [Reset = 00000000h]

MDIO\_USER\_INT\_MASK\_SET\_REG is shown in [Table 9-51](#).

Return to the [Summary Table](#).

MDIO User Interrupt Mask Set Register

**Table 9-51. MDIO\_USER\_INT\_MASK\_SET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	USERINTMASKSET	R/W	0h	MDIO user interrupt mask set

### 9.6.20 MDIO\_USER\_INT\_MASK\_CLEAR\_REG Register (Offset = F2Ch) [Reset = 00000000h]

MDIO\_USER\_INT\_MASK\_CLEAR\_REG is shown in [Table 9-52](#).

Return to the [Summary Table](#).

MDIO User Interrupt Mask Clear Register

**Table 9-52. MDIO\_USER\_INT\_MASK\_CLEAR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	USERINTMASKCLR	R/W	0h	MDIO user interrupt mask clear

### 9.6.21 MDIO\_MANUAL\_IF\_REG Register (Offset = F30h) [Reset = 00000000h]

MDIO\_MANUAL\_IF\_REG is shown in [Table 9-53](#).

Return to the [Summary Table](#).

MDIO Manual Interface Register

**Table 9-53. MDIO\_MANUAL\_IF\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2	mdio_mdclk_o	R/W	0h	MDIO Clock Output
1	mdio_oe	R/W	0h	MDIO Output Enable
0	mdio_pin	R/W	0h	MDIO Pin

### 9.6.22 MDIO\_POLL\_REG Register (Offset = F34h) [Reset = XXXXXX00h]

MDIO\_POLL\_REG is shown in [Table 9-54](#).

Return to the [Summary Table](#).

MDIO Poll Register

**Table 9-54. MDIO\_POLL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	manualmode	R/W	0h	MDIO Manual Mode
30	statechangemode	R/W	0h	MDIO State Change Mode
29-8	RESERVED	R	0h	
7-0	ipg	R/W	0h	MDIO IPG

### 9.6.23 MDIO\_POLL\_EN\_REG Register (Offset = F38h) [Reset = FFFFFFFFh]

MDIO\_POLL\_EN\_REG is shown in [Table 9-55](#).

Return to the [Summary Table](#).

MDIO Poll Enable Register

**Table 9-55. MDIO\_POLL\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	poll_en	R/W	FFFFFFFh	MDIO Poll Enable

### 9.6.24 MDIO\_CLAUS45\_REG Register (Offset = F3Ch) [Reset = 00000000h]

MDIO\_CLAUS45\_REG is shown in [Table 9-56](#).

Return to the [Summary Table](#).

MDIO Clause45 Register

**Table 9-56. MDIO\_CLAUS45\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	clause45	R/W	0h	MDIO Clause 45

### 9.6.25 MDIO\_USER\_ADDR0\_REG Register (Offset = F40h) [Reset = 00000000h]

MDIO\_USER\_ADDR0\_REG is shown in [Table 9-57](#).

Return to the [Summary Table](#).

MDIO Address 0 Register

**Table 9-57. MDIO\_USER\_ADDR0\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	user_addr0	R/W	0h	MDIO USER Address 0

### 9.6.26 MDIO\_USER\_ADDR1\_REG Register (Offset = F44h) [Reset = 00000000h]

MDIO\_USER\_ADDR1\_REG is shown in [Table 9-58](#).

Return to the [Summary Table](#).

MDIO Address 1 Register

**Table 9-58. MDIO\_USER\_ADDR1\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	user_addr1	R/W	0h	MDIO USER Address 1



### 9.6.27 USER\_GROUP0\_USER\_ACCESS\_REG Register (Offset = F80h) [Reset = XX000000h]

USER\_GROUP0\_USER\_ACCESS\_REG is shown in [Table 9-59](#).

Return to the [Summary Table](#).

MDIO User Access Register

**Table 9-59. USER\_GROUP0\_USER\_ACCESS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	GO	R/W	0h	Go
30	WRITE	R/W	0h	Write
29	ACK	R/W	0h	Acknowledge
28-26	RESERVED	R	0h	
25-21	REGADR	R/W	0h	Register address
20-16	PHYADR	R/W	0h	PHY address
15-0	DATA	R/W	0h	User data

### 9.6.28 USER\_GROUP0\_USER\_PHY\_SEL\_REG Register (Offset = F84h) [Reset = 000000X0h]

USER\_GROUP0\_USER\_PHY\_SEL\_REG is shown in [Table 9-60](#).

Return to the [Summary Table](#).

MDIO User PHY Select Register

**Table 9-60. USER\_GROUP0\_USER\_PHY\_SEL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	LINKSEL	R/W	0h	Link status determination select
6	LINKINT_ENABLE	R/W	0h	Link change interrupt enable
5	RESERVED	R	0h	
4-0	PHYADR_MON	R/W	0h	PHY address whose link status is monitored

### 9.6.29 USER\_GROUP1\_USER\_ACCESS\_REG Register (Offset = F88h) [Reset = XX000000h]

USER\_GROUP1\_USER\_ACCESS\_REG is shown in [Table 9-61](#).

Return to the [Summary Table](#).

MDIO User Access Register

**Table 9-61. USER\_GROUP1\_USER\_ACCESS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	GO	R/W	0h	Go
30	WRITE	R/W	0h	Write
29	ACK	R/W	0h	Acknowledge
28-26	RESERVED	R	0h	
25-21	REGADR	R/W	0h	Register address
20-16	PHYADR	R/W	0h	PHY address
15-0	DATA	R/W	0h	User data

### 9.6.30 USER\_GROUP1\_USER\_PHY\_SEL\_REG Register (Offset = F8Ch) [Reset = 000000X0h]

USER\_GROUP1\_USER\_PHY\_SEL\_REG is shown in [Table 9-62](#).

Return to the [Summary Table](#).

MDIO User PHY Select Register

**Table 9-62. USER\_GROUP1\_USER\_PHY\_SEL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	LINKSEL	R/W	0h	Link status determination select
6	LINKINT_ENABLE	R/W	0h	Link change interrupt enable
5	RESERVED	R	0h	
4-0	PHYADR_MON	R/W	0h	PHY address whose link status is monitored

### 9.6.31 REGS\_INT\_SS\_C0\_TH\_THRESH\_PULSE\_EN\_REG Register (Offset = 1800h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_TH\_THRESH\_PULSE\_EN\_REG is shown in [Table 9-63](#).

Return to the [Summary Table](#).

Core 0 THost Threshold Pulse Interrupt Enable Register

**Table 9-63. REGS\_INT\_SS\_C0\_TH\_THRESH\_PULSE\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH_THRESH_PULSE_EN	R/W	0h	THost Threshold Pulse Interrupt Enable Register

### 9.6.32 REGS\_INT\_SS\_C0\_TH\_PULSE\_EN\_REG Register (Offset = 1804h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_TH\_PULSE\_EN\_REG is shown in [Table 9-64](#).

Return to the [Summary Table](#).

Core 0 THost Pulse Interrupt Enable Register

**Table 9-64. REGS\_INT\_SS\_C0\_TH\_PULSE\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH_PULSE_EN	R/W	0h	Core 0 THost Pulse Interrupt Enable Register

### 9.6.33 REGS\_INT\_SS\_C0\_FH\_PULSE\_EN\_REG Register (Offset = 1808h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_FH\_PULSE\_EN\_REG is shown in [Table 9-65](#).

Return to the [Summary Table](#).

Core 0 FHost Pulse Interrupt Enable Register

**Table 9-65. REGS\_INT\_SS\_C0\_FH\_PULSE\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	FH_PULSE_EN	R/W	0h	Core 0 FHost Pulse Interrupt Enable Register

### 9.6.34 REGS\_INT\_SS\_C0\_MISC\_EN\_REG Register (Offset = 180Ch) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_MISC\_EN\_REG is shown in [Table 9-66](#).

Return to the [Summary Table](#).

Core 0 Misc Interrupt Enable Register

**Table 9-66. REGS\_INT\_SS\_C0\_MISC\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	DED_PEND_EN	R/W	0h	Core 0 MISC DED Memory Protect Error Interrupt Enable
5	SEC_PEND_EN	R/W	0h	Core 0 MISC SEC Memory Protect Error Interrupt Enable
4	EVNT_PEND_EN	R/W	0h	Core 0 MISC CPTS Event Interrupt Enable
3	STAT_PEND_EN	R/W	0h	Core 0 MISC Statistics Interrupt Enable - OR of bits n downto 0
2	HOST_PEND_EN	R/W	0h	Core 0 MISC Host Interrupt Enable
1	MDIO_LINKINT_EN	R/W	0h	Core 0 MISC MDIO linkint - OR of bits 1 and 0
0	MDIO_USERINT_EN	R/W	0h	Core 0 MISC_MDIO userint interrupt enable - OR of bits 1 and 0



### 9.6.35 REGS\_INT\_SS\_C0\_TH\_THRESH\_PULSE\_STATUS\_REG Register (Offset = 1810h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_TH\_THRESH\_PULSE\_STATUS\_REG is shown in [Table 9-67](#).

Return to the [Summary Table](#).

THost Threshold Pulse Interrupt Status Register

**Table 9-67. REGS\_INT\_SS\_C0\_TH\_THRESH\_PULSE\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH_THRESH_PULSE_STATUS	R	0h	Core 0 THost Threshold Pulse Interrupt Status Register

### 9.6.36 REGS\_INT\_SS\_C0\_TH\_PULSE\_STATUS\_REG Register (Offset = 1814h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_TH\_PULSE\_STATUS\_REG is shown in [Table 9-68](#).

Return to the [Summary Table](#).

THost Pulse Interrupt Status Register

**Table 9-68. REGS\_INT\_SS\_C0\_TH\_PULSE\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH_PULSE_STATUS	R	0h	Core 0 THost Pulse Interrupt Status Register

### 9.6.37 REGS\_INT\_SS\_C0\_FH\_PULSE\_STATUS\_REG Register (Offset = 1818h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_FH\_PULSE\_STATUS\_REG is shown in [Table 9-69](#).

Return to the [Summary Table](#).

FHost Pulse Interrupt Status Register

**Table 9-69. REGS\_INT\_SS\_C0\_FH\_PULSE\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	FH_PULSE_STATUS	R	0h	Core 0 FHost Pulse Interrupt Status Register

### 9.6.38 REGS\_INT\_SS\_C0\_MISC\_STATUS\_REG Register (Offset = 181Ch) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_MISC\_STATUS\_REG is shown in [Table 9-70](#).

Return to the [Summary Table](#).

Misc Interrupt Status Register - Set bits in this register indicate that an enabled interrupt is asserted

**Table 9-70. REGS\_INT\_SS\_C0\_MISC\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	DED_PEND	R/W	0h	Core 0 MISC DED Memory Protect Error Interrupt
5	SEC_PEND	R/W	0h	Core 0 MISC SEC Memory Protect Error Interrupt
4	EVNT_PEND	R/W	0h	Core 0 MISC CPTS Event Interrupt
3	STAT_PEND	R/W	0h	Core 0 MISC Statistics Interrupt - OR of bits n downto 0
2	HOST_PEND	R/W	0h	Core 0 MISC Host Interrupt Enable
1	MDIO_LINKINT	R/W	0h	Core 0 MISC MDIO linkint - OR of bits 1 and 0
0	MDIO_USERINT	R/W	0h	Core 0 MISC_MDIO userint interrupt - OR of bits 1 and 0

### 9.6.39 REGS\_INT\_SS\_C0\_TH\_IMAX\_REG Register (Offset = 1820h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_TH\_IMAX\_REG is shown in [Table 9-71](#).

Return to the [Summary Table](#).

Core 0 THost Interrupt Max Register Register

**Table 9-71. REGS\_INT\_SS\_C0\_TH\_IMAX\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	TH_IMAX	R/W	0h	Core 0 THost Interrupt Max Register Register

#### 9.6.40 REGS\_INT\_SS\_C0\_FH\_IMAX\_REG Register (Offset = 1824h) [Reset = 00000000h]

REGS\_INT\_SS\_C0\_FH\_IMAX\_REG is shown in [Table 9-72](#).

Return to the [Summary Table](#).

Core 0 FHost Interrupt Max Register Register

**Table 9-72. REGS\_INT\_SS\_C0\_FH\_IMAX\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R	0h	
5-0	FH_IMAX	R/W	0h	Core 0 FHost Interrupt Max Register Register

### 9.6.41 CPSW\_NC\_CPSW\_ID\_VER\_REG Register (Offset = 00020000h) [Reset = 6B901903h]

CPSW\_NC\_CPSW\_ID\_VER\_REG is shown in [Table 9-73](#).

Return to the [Summary Table](#).

CPSW ID Version

**Table 9-73. CPSW\_NC\_CPSW\_ID\_VER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	IDENT	R	6B90h	Identification Value
15-11	RTL_VER	R	3h	RTL Version Value
10-8	MAJOR_VER	R	1h	Major Version Value
7-0	MINOR_VER	R	3h	Minor Version Value

### 9.6.42 CPSW\_NC\_CONTROL\_REG Register (Offset = 00020004h) [Reset = XXXX0000h]

CPSW\_NC\_CONTROL\_REG is shown in [Table 9-74](#).

Return to the [Summary Table](#).

CPSW Switch Control

**Table 9-74. CPSW\_NC\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	ECC_CRC_MODE	R/W	0h	ECC CRC Mode
30-19	RESERVED	R	0h	
18	EST_ENABLE	R/W	0h	Interspersed Express Traffic enable
17	RESERVED	R	0h	RESERVED
16	EEE_ENABLE	R/W	0h	Energy Efficient Ethernet enable
15	P0_RX_PASS_CRC_ERR	R/W	0h	Port 0 Pass Received CRC errors
14	P0_RX_PAD	R/W	0h	Port 0 Receive Short Packet Pad
13	P0_TX_CRC_REMOVE	R/W	0h	Port 0 Transmit CRC remove
12	P0_TX_CRC_TYPE	R/W	0h	Port 0 Transmit CRC Type
11	P8_PASS_PRI_TAGGED	R/W	0h	Port 8 Pass Priority Tagged
10	P7_PASS_PRI_TAGGED	R/W	0h	Port 7 Pass Priority Tagged
9	P6_PASS_PRI_TAGGED	R/W	0h	Port 6 Pass Priority Tagged
8	P5_PASS_PRI_TAGGED	R/W	0h	Port 5 Pass Priority Tagged
7	P4_PASS_PRI_TAGGED	R/W	0h	Port 4 Pass Priority Tagged
6	P3_PASS_PRI_TAGGED	R/W	0h	Port 3 Pass Priority Tagged
5	P2_PASS_PRI_TAGGED	R/W	0h	Port 2 Pass Priority Tagged
4	P1_PASS_PRI_TAGGED	R/W	0h	Port 1 Pass Priority Tagged
3	P0_PASS_PRI_TAGGED	R/W	0h	Port 0 Pass Priority Tagged
2	P0_ENABLE	R/W	0h	Port 0 Enable
1	VLAN_AWARE	R/W	0h	VLAN Aware Mode
0	S_CN_SWITCH	R/W	0h	VLAN Aware Mode



### 9.6.43 CPSW\_NC\_EM\_CONTROL\_REG Register (Offset = 00020010h) [Reset = 00000000h]

CPSW\_NC\_EM\_CONTROL\_REG is shown in [Table 9-75](#).

Return to the [Summary Table](#).

CPSW Emulation Control

**Table 9-75. CPSW\_NC\_EM\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SOFT	R/W	0h	Emulation Soft Bit
0	FREE	R/W	0h	Emulation Free Bit

#### 9.6.44 CPSW\_NC\_STAT\_PORT\_EN\_REG Register (Offset = 00020014h) [Reset = 00000000h]

CPSW\_NC\_STAT\_PORT\_EN\_REG is shown in [Table 9-76](#).

Return to the [Summary Table](#).

CPSW Statistics Port Enable

**Table 9-76. CPSW\_NC\_STAT\_PORT\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	P8_STAT_EN	R/W	0h	Port 8 Statistics Enable
7	P7_STAT_EN	R/W	0h	Port 7 Statistics Enable
6	P6_STAT_EN	R/W	0h	Port 6 Statistics Enable
5	P5_STAT_EN	R/W	0h	Port 5 Statistics Enable
4	P4_STAT_EN	R/W	0h	Port 4 Statistics Enable
3	P3_STAT_EN	R/W	0h	Port 3 Statistics Enable
2	P2_STAT_EN	R/W	0h	Port 2 Statistics Enable
1	P1_STAT_EN	R/W	0h	Port 1 Statistics Enable
0	P0_STAT_EN	R/W	0h	Port 0 Statistics Enable

### 9.6.45 CPSW\_NC\_PTYPE\_REG Register (Offset = 00020018h) [Reset = 000000X0h]

CPSW\_NC\_PTYPE\_REG is shown in [Table 9-77](#).

Return to the [Summary Table](#).

CPSW Transmit Priority Type

**Table 9-77. CPSW\_NC\_PTYPE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RESERVED	R	0h	
16	P8_PTYPE_ESC	R/W	0h	Port 8 Priority Type Escalate
15	P7_PTYPE_ESC	R/W	0h	Port 7 Priority Type Escalate
14	P6_PTYPE_ESC	R/W	0h	Port 6 Priority Type Escalate
13	P5_PTYPE_ESC	R/W	0h	Port 5 Priority Type Escalate
12	P4_PTYPE_ESC	R/W	0h	Port 4 Priority Type Escalate
11	P3_PTYPE_ESC	R/W	0h	Port 3 Priority Type Escalate
10	P2_PTYPE_ESC	R/W	0h	Port 2 Priority Type Escalate
9	P1_PTYPE_ESC	R/W	0h	Port 1 Priority Type Escalate
8	P0_PTYPE_ESC	R/W	0h	Port 0 Priority Type Escalate
7-5	RESERVED	R	0h	
4-0	ESC_PRI_LD_VAL	R/W	0h	Escalate Priority Load Value

### 9.6.46 CPSW\_NC\_SOFT\_IDLE\_REG Register (Offset = 0002001Ch) [Reset = 00000000h]

CPSW\_NC\_SOFT\_IDLE\_REG is shown in [Table 9-78](#).

Return to the [Summary Table](#).

CPSW Software Idle

**Table 9-78. CPSW\_NC\_SOFT\_IDLE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_IDLE	R/W	0h	Software Idle

### 9.6.47 CPSW\_NC\_THRU\_RATE\_REG Register (Offset = 00020020h) [Reset = 00003XX1h]

CPSW\_NC\_THRU\_RATE\_REG is shown in [Table 9-79](#).

Return to the [Summary Table](#).

CPSW Thru Rate

**Table 9-79. CPSW\_NC\_THRU\_RATE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-12	SL_RX_THRU_RATE	R/W	3h	Switch FIFO receive through rate
11-4	RESERVED	R	0h	
3-0	P0_RX_THRU_RATE	R/W	1h	CPPI FIFO receive through rate

### 9.6.48 CPSW\_NC\_GAP\_THRESH\_REG Register (Offset = 00020024h) [Reset = 0000000Bh]

CPSW\_NC\_GAP\_THRESH\_REG is shown in [Table 9-80](#).

Return to the [Summary Table](#).

CPSW Transmit FIFO Short Gap Threshold

**Table 9-80. CPSW\_NC\_GAP\_THRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	GAP_THRESH	R/W	Bh	Short Gap Threshold

### 9.6.49 CPSW\_NC\_EEE\_PRESCALE\_REG Register (Offset = 0002002Ch) [Reset = 00000000h]

CPSW\_NC\_EEE\_PRESCALE\_REG is shown in [Table 9-81](#).

Return to the [Summary Table](#).

CPSW Energy Efficient Ethernet Prescale Value

**Table 9-81. CPSW\_NC\_EEE\_PRESCALE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11-0	EEE_PRESCALE	R/W	0h	Energy Efficient Ethernet Pre-scale count load value

### 9.6.50 CPSW\_NC\_TX\_G\_OFLOW\_THRESH\_SET\_REG Register (Offset = 00020030h) [Reset = FFFFFFFFh]

CPSW\_NC\_TX\_G\_OFLOW\_THRESH\_SET\_REG is shown in [Table 9-82](#).

Return to the [Summary Table](#).

CPSW PFC Tx Global Out Flow Threshold Set

**Table 9-82. CPSW\_NC\_TX\_G\_OFLOW\_THRESH\_SET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	PRI7	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 7
27-24	PRI6	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 6
23-20	PRI5	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 5
19-16	PRI4	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 4
15-12	PRI3	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 3
11-8	PRI2	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 2
7-4	PRI1	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 1
3-0	PRI0	R/W	Fh	Priority Based Flow Control Global Outflow Usage Threshold for Pri 0



### 9.6.51 CPSW\_NC\_TX\_G\_OFLOW\_THRESH\_CLR\_REG Register (Offset = 00020034h) [Reset = 00000000h]

CPSW\_NC\_TX\_G\_OFLOW\_THRESH\_CLR\_REG is shown in [Table 9-83](#).

Return to the [Summary Table](#).

CPSW PFC Tx Global Out Flow Threshold Clear

**Table 9-83. CPSW\_NC\_TX\_G\_OFLOW\_THRESH\_CLR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	PRI7	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 7
27-24	PRI6	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 6
23-20	PRI5	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 5
19-16	PRI4	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 4
15-12	PRI3	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 3
11-8	PRI2	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 2
7-4	PRI1	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 1
3-0	PRI0	R/W	0h	Priority Based Flow Control Global Outflow Usage Threshold for Pri 0

### 9.6.52 CPSW\_NC\_TX\_G\_BUF\_THRESH\_SET\_L\_REG Register (Offset = 00020038h) [Reset = FFFFFFFFh]

CPSW\_NC\_TX\_G\_BUF\_THRESH\_SET\_L\_REG is shown in [Table 9-84](#).

Return to the [Summary Table](#).

CPSW PFC Global Tx Buffer Threshold Set Low

**Table 9-84. CPSW\_NC\_TX\_G\_BUF\_THRESH\_SET\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PRI3	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 3
23-16	PRI2	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 2
15-8	PRI1	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 1
7-0	PRI0	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 0

### 9.6.53 CPSW\_NC\_TX\_G\_BUF\_THRESH\_SET\_H\_REG Register (Offset = 0002003Ch) [Reset = FFFFFFFFh]

CPSW\_NC\_TX\_G\_BUF\_THRESH\_SET\_H\_REG is shown in [Table 9-85](#).

Return to the [Summary Table](#).

CPSW PFC Global Tx Buffer Threshold Set High

**Table 9-85. CPSW\_NC\_TX\_G\_BUF\_THRESH\_SET\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PRI7	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 7
23-16	PRI6	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 6
15-8	PRI5	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 5
7-0	PRI4	R/W	FFh	Priority Based Flow Control Global Buffer Usage Threshold for Priority 4

### 9.6.54 CPSW\_NC\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG Register (Offset = 00020040h) [Reset = 00000000h]

CPSW\_NC\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG is shown in [Table 9-86](#).

Return to the [Summary Table](#).

CPSW PFC Global Tx Buffer Threshold Clear Low

**Table 9-86. CPSW\_NC\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PRI3	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 3
23-16	PRI2	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 2
15-8	PRI1	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 1
7-0	PRI0	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 0

### 9.6.55 CPSW\_NC\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG Register (Offset = 00020044h) [Reset = 00000000h]

CPSW\_NC\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG is shown in [Table 9-87](#).

Return to the [Summary Table](#).

CPSW PFC Global Tx Buffer Threshold Clear High

**Table 9-87. CPSW\_NC\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PRI7	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 7
23-16	PRI6	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 6
15-8	PRI5	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 5
7-0	PRI4	R/W	0h	Priority Based Flow Control Global Buffer Usage Threshold for Priority 4

### 9.6.56 CPSW\_NC\_VLAN\_LTYPE\_REG Register (Offset = 00020050h) [Reset = 88A88100h]

CPSW\_NC\_VLAN\_LTYPE\_REG is shown in [Table 9-88](#).

Return to the [Summary Table](#).

VLAN Length/type

**Table 9-88. CPSW\_NC\_VLAN\_LTYPE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	VLAN_LTYPE_OUTER	R/W	88A8h	Outer VLAN LType
15-0	VLAN_LTYPE_INNER	R/W	8100h	Inner VLAN LType

### 9.6.57 CPSW\_NC\_EST\_TS\_DOMAIN\_REG Register (Offset = 00020054h) [Reset = 00000000h]

CPSW\_NC\_EST\_TS\_DOMAIN\_REG is shown in [Table 9-89](#).

Return to the [Summary Table](#).

Enhanced Scheduled Traffic Host Event Domain

**Table 9-89. CPSW\_NC\_EST\_TS\_DOMAIN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	EST_TS_DOMAIN	R/W	0h	Enhanced Scheduled Traffic Host Event Domain

### 9.6.58 CPSW\_NC\_TX\_PRI0\_MAXLEN\_REG Register (Offset = 00020100h) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI0\_MAXLEN\_REG is shown in [Table 9-90](#).

Return to the [Summary Table](#).

Transmit Priority 0 Maximum Length

**Table 9-90. CPSW\_NC\_TX\_PRI0\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI0_MAXLEN	R/W	7E8h	Transmit Priority 0 Maximum Length



### 9.6.59 CPSW\_NC\_TX\_PRI1\_MAXLEN\_REG Register (Offset = 00020104h) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI1\_MAXLEN\_REG is shown in [Table 9-91](#).

Return to the [Summary Table](#).

Transmit Priority 1 Maximum Length

**Table 9-91. CPSW\_NC\_TX\_PRI1\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI1_MAXLEN	R/W	7E8h	Transmit Priority 1 Maximum Length

### 9.6.60 CPSW\_NC\_TX\_PRI2\_MAXLEN\_REG Register (Offset = 00020108h) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI2\_MAXLEN\_REG is shown in [Table 9-92](#).

Return to the [Summary Table](#).

Transmit Priority 2 Maximum Length

**Table 9-92. CPSW\_NC\_TX\_PRI2\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI2_MAXLEN	R/W	7E8h	Transmit Priority 2 Maximum Length

### 9.6.61 CPSW\_NC\_TX\_PRI3\_MAXLEN\_REG Register (Offset = 0002010Ch) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI3\_MAXLEN\_REG is shown in [Table 9-93](#).

Return to the [Summary Table](#).

Transmit Priority 3 Maximum Length

**Table 9-93. CPSW\_NC\_TX\_PRI3\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI3_MAXLEN	R/W	7E8h	Transmit Priority 3 Maximum Length

### 9.6.62 CPSW\_NC\_TX\_PRI4\_MAXLEN\_REG Register (Offset = 00020110h) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI4\_MAXLEN\_REG is shown in [Table 9-94](#).

Return to the [Summary Table](#).

Transmit Priority 4 Maximum Length

**Table 9-94. CPSW\_NC\_TX\_PRI4\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI4_MAXLEN	R/W	7E8h	Transmit Priority 4 Maximum Length

### 9.6.63 CPSW\_NC\_TX\_PRI5\_MAXLEN\_REG Register (Offset = 00020114h) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI5\_MAXLEN\_REG is shown in [Table 9-95](#).

Return to the [Summary Table](#).

Transmit Priority 5 Maximum Length

**Table 9-95. CPSW\_NC\_TX\_PRI5\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI5_MAXLEN	R/W	7E8h	Transmit Priority 5 Maximum Length

### 9.6.64 CPSW\_NC\_TX\_PRI6\_MAXLEN\_REG Register (Offset = 00020118h) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI6\_MAXLEN\_REG is shown in [Table 9-96](#).

Return to the [Summary Table](#).

Transmit Priority 6 Maximum Length

**Table 9-96. CPSW\_NC\_TX\_PRI6\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI6_MAXLEN	R/W	7E8h	Transmit Priority 6 Maximum Length

### 9.6.65 CPSW\_NC\_TX\_PRI7\_MAXLEN\_REG Register (Offset = 0002011Ch) [Reset = 000007E8h]

CPSW\_NC\_TX\_PRI7\_MAXLEN\_REG is shown in [Table 9-97](#).

Return to the [Summary Table](#).

Transmit Priority 7 Maximum Length

**Table 9-97. CPSW\_NC\_TX\_PRI7\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	TX_PRI7_MAXLEN	R/W	7E8h	Transmit Priority 7 Maximum Length

### 9.6.66 CPSW\_NC\_CPPI\_P0\_CONTROL\_REG Register (Offset = 00021004h) [Reset = 0000XXX0h]

CPSW\_NC\_CPPI\_P0\_CONTROL\_REG is shown in [Table 9-98](#).

Return to the [Summary Table](#).

CPPI Port 0 Control

**Table 9-98. CPSW\_NC\_CPPI\_P0\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	RX_REMAP_DSCP_V6	R/W	0h	Port 0 Remap DSCP_V6 Enable
17	RX_REMAP_DSCP_V4	R/W	0h	Port 0 Remap DSCP_V4 Enable
16	RX_REMAP_VLAN	R/W	0h	Port 0 Remap VLAN Enable
15	RX_ECC_ERR_EN	R/W	0h	Port 0 Receive ECC Error Enable
14	TX_ECC_ERR_EN	R/W	0h	Port 0 Transmit ECC Error Enable
13-4	RESERVED	R	0h	
3	TX_CHECKSUM_EN	R/W	0h	Port 0 Transmit Checksum Enable
2	DSCP_IPV6_EN	R/W	0h	Port 0 IPv6 DSCP enable
1	DSCP_IPV4_EN	R/W	0h	Port 0 IPv4 DSCP enable
0	RX_CHECKSUM_EN	R/W	0h	Port 0 Receive Checksum Enable



### 9.6.67 CPSW\_NC\_CPPI\_P0\_FLOW\_ID\_OFFSET\_REG Register (Offset = 00021008h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_FLOW\_ID\_OFFSET\_REG is shown in [Table 9-99](#).

Return to the [Summary Table](#).

CPPI Port 0 Flow ID Offset

**Table 9-99. CPSW\_NC\_CPPI\_P0\_FLOW\_ID\_OFFSET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	VALUE	R/W	0h	This value is added to the thread/Flow_ID in CPPI transmit PSI Info Word 0

### 9.6.68 CPSW\_NC\_CPPI\_P0\_BLK\_CNT\_REG Register (Offset = 00021010h) [Reset = 000000X1h]

CPSW\_NC\_CPPI\_P0\_BLK\_CNT\_REG is shown in [Table 9-100](#).

Return to the [Summary Table](#).

CPPI Port 0 FIFO Block Usage Count

**Table 9-100. CPSW\_NC\_CPPI\_P0\_BLK\_CNT\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R	0h	
12-8	TX_BLK_CNT	R	0h	Port 0 Transmit Block Count Usage
7-6	RESERVED	R	0h	
5-0	RX_BLK_CNT	R	1h	Port 0 Receive Block Count Usage

### 9.6.69 CPSW\_NC\_CPPI\_P0\_PORT\_VLAN\_REG Register (Offset = 00021014h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PORT\_VLAN\_REG is shown in [Table 9-101](#).

Return to the [Summary Table](#).

CPPI Port 0 VLAN

**Table 9-101. CPSW\_NC\_CPPI\_P0\_PORT\_VLAN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-13	PORT_PRI	R/W	0h	Port VLAN Priority
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

### 9.6.70 CPSW\_NC\_CPPI\_P0\_TX\_PRI\_MAP\_REG Register (Offset = 00021018h) [Reset = 7XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_TX\_PRI\_MAP\_REG is shown in [Table 9-102](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx Header Pri to Switch Pri Mapping

**Table 9-102. CPSW\_NC\_CPPI\_P0\_TX\_PRI\_MAP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	7h	Priority 7
27	RESERVED	R	0h	
26-24	PRI6	R/W	6h	Priority 6
23	RESERVED	R	0h	
22-20	PRI5	R/W	5h	Priority 5
19	RESERVED	R	0h	
18-16	PRI4	R/W	4h	Priority 4
15	RESERVED	R	0h	
14-12	PRI3	R/W	3h	Priority 3
11	RESERVED	R	0h	
10-8	PRI2	R/W	2h	Priority 2
7	RESERVED	R	0h	
6-4	PRI1	R/W	1h	Priority 1
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0

### 9.6.71 CPSW\_NC\_CPPI\_P0\_PRI\_CTL\_REG Register (Offset = 0002101Ch) [Reset = 0000XXXXh]

CPSW\_NC\_CPPI\_P0\_PRI\_CTL\_REG is shown in [Table 9-103](#).

Return to the [Summary Table](#).

CPPI Port 0 Priority Control

**Table 9-103. CPSW\_NC\_CPPI\_P0\_PRI\_CTL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-16	RX_FLOW_PRI	R/W	0h	Receive Priority Based Flow Control Enable (per priority)
15-9	RESERVED	R	0h	
8	RX_PTYPE	R/W	0h	Receive Priority Type
7-0	RESERVED	R	0h	

### 9.6.72 CPSW\_NC\_CPPI\_P0\_RX\_PRI\_MAP\_REG Register (Offset = 00021020h) [Reset = 7XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_PRI\_MAP\_REG is shown in [Table 9-104](#).

Return to the [Summary Table](#).

CPPI Port 0 RX Pkt Pri to Header Pri Map

**Table 9-104. CPSW\_NC\_CPPI\_P0\_RX\_PRI\_MAP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	7h	Priority 7
27	RESERVED	R	0h	
26-24	PRI6	R/W	6h	Priority 6
23	RESERVED	R	0h	
22-20	PRI5	R/W	5h	Priority 5
19	RESERVED	R	0h	
18-16	PRI4	R/W	4h	Priority 4
15	RESERVED	R	0h	
14-12	PRI3	R/W	3h	Priority 3
11	RESERVED	R	0h	
10-8	PRI2	R/W	2h	Priority 2
7	RESERVED	R	0h	
6-4	PRI1	R/W	1h	Priority 1
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0

### 9.6.73 CPSW\_NC\_CPPI\_P0\_RX\_MAXLEN\_REG Register (Offset = 00021024h) [Reset = 000005EEh]

CPSW\_NC\_CPPI\_P0\_RX\_MAXLEN\_REG is shown in [Table 9-105](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive Frame Max Length

**Table 9-105. CPSW\_NC\_CPPI\_P0\_RX\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	RX_MAXLEN	R/W	5EEh	Rx Maximum Frame Length

### 9.6.74 CPSW\_NC\_CPPI\_P0\_TX\_BLKs\_PRI\_REG Register (Offset = 00021028h) [Reset = 01245678h]

CPSW\_NC\_CPPI\_P0\_TX\_BLKs\_PRI\_REG is shown in [Table 9-106](#).

Return to the [Summary Table](#).

CPPI Port 0 Transmit Block Sub Per Priority

**Table 9-106. CPSW\_NC\_CPPI\_P0\_TX\_BLKs\_PRI\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	PRI7	R/W	0h	Priority 7 Port Transmit Blocks
27-24	PRI6	R/W	1h	Priority 6 Port Transmit Blocks
23-20	PRI5	R/W	2h	Priority 5 Port Transmit Blocks
19-16	PRI4	R/W	4h	Priority 4 Port Transmit Blocks
15-12	PRI3	R/W	5h	Priority 3 Port Transmit Blocks
11-8	PRI2	R/W	6h	Priority 2 Port Transmit Blocks
7-4	PRI1	R/W	7h	Priority 1 Port Transmit Blocks
3-0	PRI0	R/W	8h	Priority 0 Port Transmit Blocks



### 9.6.75 CPSW\_NC\_CPPI\_P0\_IDLE2LPI\_REG Register (Offset = 00021030h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_IDLE2LPI\_REG is shown in [Table 9-107](#).

Return to the [Summary Table](#).

Port 0 EEE Idle to LPI counter

**Table 9-107. CPSW\_NC\_CPPI\_P0\_IDLE2LPI\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	COUNT	R/W	0h	Port 0 EEE Idle to LPI counter load value

### 9.6.76 CPSW\_NC\_CPPI\_P0\_LPI2WAKE\_REG Register (Offset = 00021034h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_LPI2WAKE\_REG is shown in [Table 9-108](#).

Return to the [Summary Table](#).

Port 0 EEE LPI to wake counter

**Table 9-108. CPSW\_NC\_CPPI\_P0\_LPI2WAKE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	COUNT	R/W	0h	Port 0 EEE LPI to wake counter load value

### 9.6.77 CPSW\_NC\_CPPI\_P0\_EEE\_STATUS\_REG Register (Offset = 00021038h) [Reset = 00000060h]

CPSW\_NC\_CPPI\_P0\_EEE\_STATUS\_REG is shown in [Table 9-109](#).

Return to the [Summary Table](#).

Port 0 EEE status

**Table 9-109. CPSW\_NC\_CPPI\_P0\_EEE\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	TX_FIFO_EMPTY	R	1h	CPPI port 0 transmit FIFO (switch egress) is empty - contains no packets
5	RX_FIFO_EMPTY	R	1h	CPPI port 0 receive FIFO (switch ingress) is empty - contains no packets
4	TX_FIFO_HOLD	R	0h	CPPI port 0 transmit FIFO hold - asserted in the LPI state and during the LPI2WAKE count time
3	TX_WAKE	R	0h	CPPI port 0 transmit wakeup - asserted in the transmit LPI2WAKE count time
2	TX_LPI	R	0h	CPPI port 0 transmit LPI state - asserted when the port 0 transmit is in the LPI state
1	RX_LPI	R	0h	CPPI port 0 receive LPI state - asserted when the port 0 receive is in the LPI state
0	WAIT_IDLE2LPI	R	0h	CPPI port 0 wait idle to LPI - asserted when port 0 is counting the IDLE2LPI time

### 9.6.78 CPSW\_NC\_CPPI\_P0\_RX\_PKTS\_PRI\_REG Register (Offset = 0002103Ch) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_RX\_PKTS\_PRI\_REG is shown in [Table 9-110](#).

Return to the [Summary Table](#).

CPPI Port Receive Packets per priority

**Table 9-110. CPSW\_NC\_CPPI\_P0\_RX\_PKTS\_PRI\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	PRI7	R/W	0h	Priority 7 Port Port 0 Receive Packets
27-24	PRI6	R/W	0h	Priority 6 Port Port 0 Receive Packets
23-20	PRI5	R/W	0h	Priority 5 Port Port 0 Receive Packets
19-16	PRI4	R/W	0h	Priority 4 Port Port 0 Receive Packets
15-12	PRI3	R/W	0h	Priority 3 Port Port 0 Receive Packets
11-8	PRI2	R/W	0h	Priority 2 Port Port 0 Receive Packets
7-4	PRI1	R/W	0h	Priority 1 Port Port 0 Receive Packets
3-0	PRI0	R/W	0h	Priority 0 Port Port 0 Receive Packets

### 9.6.79 CPSW\_NC\_CPPI\_P0\_RX\_GAP\_REG Register (Offset = 0002104Ch) [Reset = 0100XX00h]

CPSW\_NC\_CPPI\_P0\_RX\_GAP\_REG is shown in [Table 9-111](#).

Return to the [Summary Table](#).

Port 0 Receive Gap Register

**Table 9-111. CPSW\_NC\_CPPI\_P0\_RX\_GAP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-16	RX_GAP_CNT	R/W	100h	Port 0 Receive Gap Count
15-8	RESERVED	R	0h	
7-0	RX_GAP_EN	R/W	0h	Port 0 Receive Gap Enable

### 9.6.80 CPSW\_NC\_CPPI\_P0\_FIFO\_STATUS\_REG Register (Offset = 00021050h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_FIFO\_STATUS\_REG is shown in [Table 9-112](#).

Return to the [Summary Table](#).

Port 0 FIFO Status

**Table 9-112. CPSW\_NC\_CPPI\_P0\_FIFO\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	tx_pri_active	R	0h	Port 0 FIFO Status

### 9.6.81 CPSW\_NC\_CPPI\_P0\_MAX\_BLKs\_REG Register (Offset = 00021080h) [Reset = 00001004h]

CPSW\_NC\_CPPI\_P0\_MAX\_BLKs\_REG is shown in [Table 9-113](#).

Return to the [Summary Table](#).

Port 0 FIFO Max Blocks

**Table 9-113. CPSW\_NC\_CPPI\_P0\_MAX\_BLKs\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	TX_MAX_BLKs	R/W	10h	Transmit FIFO maximum blocks
7-0	RX_MAX_BLKs	R/W	4h	Receive FIFO maximum blocks

### 9.6.82 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_0 Register (Offset = 00021120h) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_0 is shown in [Table 9-114](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-114. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority



### 9.6.83 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_1 Register (Offset = 00021124h) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_1 is shown in [Table 9-115](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-115. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.84 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_2 Register (Offset = 00021128h) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_2 is shown in [Table 9-116](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-116. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.85 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_3 Register (Offset = 0002112Ch) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_3 is shown in [Table 9-117](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-117. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.86 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_4 Register (Offset = 00021130h) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_4 is shown in [Table 9-118](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-118. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.87 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_5 Register (Offset = 00021134h) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_5 is shown in [Table 9-119](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-119. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.88 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_6 Register (Offset = 00021138h) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_6 is shown in [Table 9-120](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-120. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.89 CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_7 Register (Offset = 0002113Ch) [Reset = 0XXXXXXXh]

CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_7 is shown in [Table 9-121](#).

Return to the [Summary Table](#).

CPPI Port 0 Receive IPV4/IPV6 DSCP Map N

**Table 9-121. CPSW\_NC\_CPPI\_P0\_RX\_DSCP\_MAP\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.90 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_0 Register (Offset = 00021140h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_0 is shown in [Table 9-122](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-122. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR



### 9.6.91 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_1 Register (Offset = 00021144h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_1 is shown in [Table 9-123](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-123. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.92 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_2 Register (Offset = 00021148h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_2 is shown in [Table 9-124](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-124. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.93 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_3 Register (Offset = 0002114Ch) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_3 is shown in [Table 9-125](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-125. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.94 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_4 Register (Offset = 00021150h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_4 is shown in [Table 9-126](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-126. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.95 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_5 Register (Offset = 00021154h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_5 is shown in [Table 9-127](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-127. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.96 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_6 Register (Offset = 00021158h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_6 is shown in [Table 9-128](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-128. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.97 CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_7 Register (Offset = 0002115Ch) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_7 is shown in [Table 9-129](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Committed Information Rate

**Table 9-129. CPSW\_NC\_CPPI\_P0\_PRI\_CIR\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N CIR

### 9.6.98 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_0 Register (Offset = 00021160h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_0 is shown in [Table 9-130](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-130. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR



### 9.6.99 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_1 Register (Offset = 00021164h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_1 is shown in [Table 9-131](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-131. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.100 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_2 Register (Offset = 00021168h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_2 is shown in [Table 9-132](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-132. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.101 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_3 Register (Offset = 0002116Ch) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_3 is shown in [Table 9-133](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-133. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.102 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_4 Register (Offset = 00021170h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_4 is shown in [Table 9-134](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-134. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.103 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_5 Register (Offset = 00021174h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_5 is shown in [Table 9-135](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-135. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.104 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_6 Register (Offset = 00021178h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_6 is shown in [Table 9-136](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-136. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.105 CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_7 Register (Offset = 0002117Ch) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_7 is shown in [Table 9-137](#).

Return to the [Summary Table](#).

CPPI Port 0 Rx Priority P Excess Information Rate

**Table 9-137. CPSW\_NC\_CPPI\_P0\_PRI\_EIR\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N EIR

### 9.6.106 CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_SET\_L\_REG Register (Offset = 00021180h) [Reset = 1FXFXFXh]

CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_SET\_L\_REG is shown in [Table 9-138](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Destination Threshold Set Low

**Table 9-138. CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_SET\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 0



### 9.6.107 CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_SET\_H\_REG Register (Offset = 00021184h) [Reset = 1FXFXFXFh]

CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_SET\_H\_REG is shown in [Table 9-139](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Destination Threshold Set High

**Table 9-139. CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_SET\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 4

### 9.6.108 CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_CLR\_L\_REG Register (Offset = 00021188h) [Reset = 00X0X0X0h]

CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_CLR\_L\_REG is shown in [Table 9-140](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Destination Threshold Clr Low

**Table 9-140. CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_CLR\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 0

### 9.6.109 CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_CLR\_H\_REG Register (Offset = 0002118Ch) [Reset = 00X0X0X0h]

CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_CLR\_H\_REG is shown in [Table 9-141](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Destination Threshold Clr High

**Table 9-141. CPSW\_NC\_CPPI\_P0\_TX\_D\_THRESH\_CLR\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 4

### 9.6.110 CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_SET\_L\_REG Register (Offset = 00021190h) [Reset = 1FXFXFXFh]

CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_SET\_L\_REG is shown in [Table 9-142](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Global Buffer Threshold Set Low

**Table 9-142. CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_SET\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 0

### 9.6.111 CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_SET\_H\_REG Register (Offset = 00021194h) [Reset = 1FXFXFXFh]

CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_SET\_H\_REG is shown in [Table 9-143](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Global Buffer Threshold Set High

**Table 9-143. CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_SET\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 4

### 9.6.112 CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG Register (Offset = 00021198h) [Reset = 00X0X0X0h]

CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG is shown in [Table 9-144](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Global Buffer Threshold Clr Low

**Table 9-144. CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 0

### 9.6.113 CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG Register (Offset = 0002119Ch) [Reset = 00X0X0X0h]

CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG is shown in [Table 9-145](#).

Return to the [Summary Table](#).

CPPI Port 0 Tx PFC Global Buffer Threshold Clr High

**Table 9-145. CPSW\_NC\_CPPI\_P0\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 4

### 9.6.114 CPSW\_NC\_CPPI\_P0\_SRC\_ID\_A\_REG Register (Offset = 00021300h) [Reset = 04030201h]

CPSW\_NC\_CPPI\_P0\_SRC\_ID\_A\_REG is shown in [Table 9-146](#).

Return to the [Summary Table](#).

CPPI Port 0 CPPI Source ID A

**Table 9-146. CPSW\_NC\_CPPI\_P0\_SRC\_ID\_A\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PORT4	R/W	4h	Port 4 CPPI Info Word0 Source ID Value
23-16	PORT3	R/W	3h	Port 3 CPPI Info Word0 Source ID Value
15-8	PORT2	R/W	2h	Port 2 CPPI Info Word0 Source ID Value
7-0	PORT1	R/W	1h	Port 1 CPPI Info Word0 Source ID Value



### 9.6.115 CPSW\_NC\_CPPI\_P0\_SRC\_ID\_B\_REG Register (Offset = 00021304h) [Reset = 08070605h]

CPSW\_NC\_CPPI\_P0\_SRC\_ID\_B\_REG is shown in [Table 9-147](#).

Return to the [Summary Table](#).

CPPI Port 0 CPPI Source ID B

**Table 9-147. CPSW\_NC\_CPPI\_P0\_SRC\_ID\_B\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PORT8	R/W	8h	Port 8 CPPI Info Word0 Source ID Value
23-16	PORT7	R/W	7h	Port 7 CPPI Info Word0 Source ID Value
15-8	PORT6	R/W	6h	Port 6 CPPI Info Word0 Source ID Value
7-0	PORT5	R/W	5h	Port 5 CPPI Info Word0 Source ID Value

### 9.6.116 CPSW\_NC\_CPPI\_P0\_HOST\_BLKs\_PRI\_REG Register (Offset = 00021320h) [Reset = 00000000h]

CPSW\_NC\_CPPI\_P0\_HOST\_BLKs\_PRI\_REG is shown in [Table 9-148](#).

Return to the [Summary Table](#).

CPPI Port 0 Host Blocks Priority

**Table 9-148. CPSW\_NC\_CPPI\_P0\_HOST\_BLKs\_PRI\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	PRI7	R/W	0h	Priority 7 Host Blocks
27-24	PRI6	R/W	0h	Priority 6 Host Blocks
23-20	PRI5	R/W	0h	Priority 5 Host Blocks
19-16	PRI4	R/W	0h	Priority 4 Host Blocks
15-12	PRI3	R/W	0h	Priority 3 Host Blocks
11-8	PRI2	R/W	0h	Priority 2 Host Blocks
7-4	PRI1	R/W	0h	Priority 1 Host Blocks
3-0	PRI0	R/W	0h	Priority 0 Host Blocks

### 9.6.117 CPSW\_NC\_ETH\_MAC\_0\_PN\_RESERVED\_REG Register (Offset = 00022000h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RESERVED\_REG is shown in [Table 9-149](#).

Return to the [Summary Table](#).

Reserved

**Table 9-149. CPSW\_NC\_ETH\_MAC\_0\_PN\_RESERVED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved register for memory map alignment

### 9.6.118 CPSW\_NC\_ETH\_MAC\_0\_PN\_CONTROL\_REG Register (Offset = 00022004h) [Reset = 000XXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_CONTROL\_REG is shown in [Table 9-150](#).

Return to the [Summary Table](#).

Enet Port N Control

**Table 9-150. CPSW\_NC\_ETH\_MAC\_0\_PN\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	0h	
17	EST_PORT_EN	R/W	0h	EST Port Enable
16	RESERVED	R	0h	
15	RX_ECC_ERR_EN	R/W	0h	Port 0 Receive ECC Error Enable
14	TX_ECC_ERR_EN	R/W	0h	Port 0 Transmit ECC Error Enable
13	RESERVED	R	0h	
12	TX_LPI_CLKSTOP_EN	R/W	0h	Transmit LPI clockstop enable
11-3	RESERVED	R	0h	
2	DSCP_IPV6_EN	R/W	0h	IPv6 DSCP enable
1	DSCP_IPV4_EN	R/W	0h	IPv4 DSCP enable
0	RESERVED	R	0h	

### 9.6.119 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAX\_BLKs\_REG Register (Offset = 00022008h) [Reset = 00001004h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAX\_BLKs\_REG is shown in [Table 9-151](#).

Return to the [Summary Table](#).

Enet Port N FIFO Max Blocks

**Table 9-151. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAX\_BLKs\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	TX_MAX_BLKs	R/W	10h	Transmit FIFO maximum blocks
7-0	RX_MAX_BLKs	R/W	4h	Receive FIFO maximum blocks

### 9.6.120 CPSW\_NC\_ETH\_MAC\_0\_PN\_BLK\_CNT\_REG Register (Offset = 00022010h) [Reset = 0000X0X1h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_BLK\_CNT\_REG is shown in [Table 9-152](#).

Return to the [Summary Table](#).

Enet Port N FIFO Block Usage Count

**Table 9-152. CPSW\_NC\_ETH\_MAC\_0\_PN\_BLK\_CNT\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-16	RX_BLK_CNT_P	R	0h	Receive Preempt Queue Block Count Usage
15-13	RESERVED	R	0h	
12-8	TX_BLK_CNT	R	0h	Transmit Block Count Usage
7-6	RESERVED	R	0h	
5-0	RX_BLK_CNT_E	R	1h	Receive Block Count Usage

### 9.6.121 CPSW\_NC\_ETH\_MAC\_0\_PN\_PORT\_VLAN\_REG Register (Offset = 00022014h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PORT\_VLAN\_REG is shown in [Table 9-153](#).

Return to the [Summary Table](#).

Enet Port N VLAN

**Table 9-153. CPSW\_NC\_ETH\_MAC\_0\_PN\_PORT\_VLAN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-13	PORT_PRI	R/W	0h	Port VLAN Priority
12	PORT_CFI	R/W	0h	Port CFI bit
11-0	PORT_VID	R/W	0h	Port VLAN ID

### 9.6.122 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_PRI\_MAP\_REG Register (Offset = 00022018h) [Reset = 7XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_PRI\_MAP\_REG is shown in [Table 9-154](#).

Return to the [Summary Table](#).

Enet Port N Tx Header Pri to Switch Pri Mapping

**Table 9-154. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_PRI\_MAP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	7h	Priority 7
27	RESERVED	R	0h	
26-24	PRI6	R/W	6h	Priority 6
23	RESERVED	R	0h	
22-20	PRI5	R/W	5h	Priority 5
19	RESERVED	R	0h	
18-16	PRI4	R/W	4h	Priority 4
15	RESERVED	R	0h	
14-12	PRI3	R/W	3h	Priority 3
11	RESERVED	R	0h	
10-8	PRI2	R/W	2h	Priority 2
7	RESERVED	R	0h	
6-4	PRI1	R/W	1h	Priority 1
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0



### 9.6.123 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CTL\_REG Register (Offset = 0002201Ch) [Reset = 00009XXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CTL\_REG is shown in [Table 9-155](#).

Return to the [Summary Table](#).

Enet Port N Priority Control

**Table 9-155. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CTL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	TX_FLOW_PRI	R/W	0h	Transmit Priority Based Flow Control Enable (per priority)
23-16	RX_FLOW_PRI	R/W	0h	Receive Priority Based Flow Control Enable (per priority)
15-12	TX_HOST_BLKs_REM	R/W	9h	Transmit FIFO Blocks that must be free before a non rate-limited CPPI Port 0 receive thread can begin sending a packet
11-0	RESERVED	R	0h	

### 9.6.124 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_PRI\_MAP\_REG Register (Offset = 00022020h) [Reset = 7XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_PRI\_MAP\_REG is shown in [Table 9-156](#).

Return to the [Summary Table](#).

Enet Port N RX Pkt Pri to Header Pri Map

**Table 9-156. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_PRI\_MAP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	7h	Priority 7
27	RESERVED	R	0h	
26-24	PRI6	R/W	6h	Priority 6
23	RESERVED	R	0h	
22-20	PRI5	R/W	5h	Priority 5
19	RESERVED	R	0h	
18-16	PRI4	R/W	4h	Priority 4
15	RESERVED	R	0h	
14-12	PRI3	R/W	3h	Priority 3
11	RESERVED	R	0h	
10-8	PRI2	R/W	2h	Priority 2
7	RESERVED	R	0h	
6-4	PRI1	R/W	1h	Priority 1
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	Priority 0

### 9.6.125 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_MAXLEN\_REG Register (Offset = 00022024h) [Reset = 000005EEh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_MAXLEN\_REG is shown in [Table 9-157](#).

Return to the [Summary Table](#).

Enet Port N Receive Frame Max Length

**Table 9-157. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_MAXLEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	0h	
13-0	RX_MAXLEN	R/W	5EEh	Rx Maximum Frame Length

### 9.6.126 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_BLKs\_PRI\_REG Register (Offset = 00022028h) [Reset = 01245678h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_BLKs\_PRI\_REG is shown in [Table 9-158](#).

Return to the [Summary Table](#).

Enet Port N Transmit Block Sub Per Priority

**Table 9-158. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_BLKs\_PRI\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	PRI7	R/W	0h	Priority 7 Port Transmit Blocks
27-24	PRI6	R/W	1h	Priority 6 Port Transmit Blocks
23-20	PRI5	R/W	2h	Priority 5 Port Transmit Blocks
19-16	PRI4	R/W	4h	Priority 4 Port Transmit Blocks
15-12	PRI3	R/W	5h	Priority 3 Port Transmit Blocks
11-8	PRI2	R/W	6h	Priority 2 Port Transmit Blocks
7-4	PRI1	R/W	7h	Priority 1 Port Transmit Blocks
3-0	PRI0	R/W	8h	Priority 0 Port Transmit Blocks

### 9.6.127 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_FLOW\_THRESH\_REG Register (Offset = 0002202Ch) [Reset = 00000040h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_FLOW\_THRESH\_REG is shown in [Table 9-159](#).

Return to the [Summary Table](#).

Enet MAC Receive Flow Threshold in Receive Buffer Words

**Table 9-159. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_FLOW\_THRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8-0	COUNT	R/W	40h	Receive Flow Threshold in Words

### 9.6.128 CPSW\_NC\_ETH\_MAC\_0\_PN\_IDLE2LPI\_REG Register (Offset = 00022030h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_IDLE2LPI\_REG is shown in [Table 9-160](#).

Return to the [Summary Table](#).

Enet Port N EEE Idle to LPI counter

**Table 9-160. CPSW\_NC\_ETH\_MAC\_0\_PN\_IDLE2LPI\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	COUNT	R/W	0h	EEE Idle to LPI counter load value

### 9.6.129 CPSW\_NC\_ETH\_MAC\_0\_PN\_LPI2WAKE\_REG Register (Offset = 00022034h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_LPI2WAKE\_REG is shown in [Table 9-161](#).

Return to the [Summary Table](#).

Enet Port N EEE LPI to wake counter

**Table 9-161. CPSW\_NC\_ETH\_MAC\_0\_PN\_LPI2WAKE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	COUNT	R/W	0h	EEE LPI to wake counter load value

### 9.6.130 CPSW\_NC\_ETH\_MAC\_0\_PN\_EEE\_STATUS\_REG Register (Offset = 00022038h) [Reset = 00000062h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_EEE\_STATUS\_REG is shown in [Table 9-162](#).

Return to the [Summary Table](#).

Enet Port N EEE status

**Table 9-162. CPSW\_NC\_ETH\_MAC\_0\_PN\_EEE\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6	TX_FIFO_EMPTY	R	1h	Transmit FIFO (switch egress) is empty - contains no packets
5	RX_FIFO_EMPTY	R	1h	Receive FIFO (switch ingress) is empty - contains no packets
4	TX_FIFO_HOLD	R	0h	Transmit FIFO hold - asserted in the LPI state and during the LPI2WAKE count time
3	TX_WAKE	R	0h	Transmit wakeup - asserted in the transmit LPI2WAKE count time
2	TX_LPI	R	0h	Transmit LPI state - asserted when the port 0 transmit is in the LPI state
1	RX_LPI	R	1h	Receive LPI state - asserted when the port 0 receive is in the LPI state
0	WAIT_IDLE2LPI	R	0h	CPPI port 0 wait idle to LPI - asserted when port 0 is counting the IDLE2LPI time



### 9.6.131 CPSW\_NC\_ETH\_MAC\_0\_PN\_FIFO\_STATUS\_REG Register (Offset = 00022050h) [Reset = 0000FF00h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_FIFO\_STATUS\_REG is shown in [Table 9-163](#).

Return to the [Summary Table](#).

Enet Port N FIFO STATUS

**Table 9-163. CPSW\_NC\_ETH\_MAC\_0\_PN\_FIFO\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R	0h	
18	est_bufact	R	0h	Transmit FIFO EST Buffer Active
17	est_add_err	R	0h	Transmit FIFO EST Address Error
16	est_cnt_err	R	0h	Transmit FIFO EST Count Error
15-8	tx_e_mac_allow	R	FFh	Transmit FIFO Express Queue Priority Allow
7-0	tx_pri_active	R	0h	Transmit FIFO Priority Active

### 9.6.132 CPSW\_NC\_ETH\_MAC\_0\_PN\_EST\_CONTROL\_REG Register (Offset = 00022060h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_EST\_CONTROL\_REG is shown in [Table 9-164](#).

Return to the [Summary Table](#).

Enet Port N EST CONTROL

**Table 9-164. CPSW\_NC\_ETH\_MAC\_0\_PN\_EST\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-16	est_fill_margin	R/W	0h	Transmit FIFO EST Fill Margin
15-9	est_preempt_comp	R/W	0h	Transmit FIFO EST Preempt Comparison Value to Clear wire
8	est_fill_en	R/W	0h	Transmit FIFO EST Fill Enable
7-5	est_ts_pri	R/W	0h	Transmit FIFO EST TimeStamp Priority
4	est_ts_onepri	R/W	0h	Transmit FIFO EST TimeStamp One Priority
3	est_ts_first	R/W	0h	Transmit FIFO EST TimeStamp First Express Packet
2	est_ts_en	R/W	0h	Transmit FIFO EST TimeStamp Enable
1	est_bufsel	R/W	0h	Transmit FIFO EST Buffer Select
0	est_onebuf	R/W	0h	Transmit FIFO EST One Buffer

### 9.6.133 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_0 Register (Offset = 00022120h) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_0 is shown in [Table 9-165](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-165. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.134 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_1 Register (Offset = 00022124h) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_1 is shown in [Table 9-166](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-166. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.135 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_2 Register (Offset = 00022128h) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_2 is shown in [Table 9-167](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-167. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.136 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_3 Register (Offset = 0002212Ch) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_3 is shown in [Table 9-168](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-168. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.137 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_4 Register (Offset = 00022130h) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_4 is shown in [Table 9-169](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-169. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.138 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_5 Register (Offset = 00022134h) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_5 is shown in [Table 9-170](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-170. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority



### 9.6.139 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_6 Register (Offset = 00022138h) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_6 is shown in [Table 9-171](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-171. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.140 CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_7 Register (Offset = 0002213Ch) [Reset = 0XXXXXXXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_7 is shown in [Table 9-172](#).

Return to the [Summary Table](#).

Enet Port N Receive IPV4/IPV6 DSCP Map M

**Table 9-172. CPSW\_NC\_ETH\_MAC\_0\_PN\_RX\_DSCP\_MAP\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-28	PRI7	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+7 is mapped to this received priority
27	RESERVED	R	0h	
26-24	PRI6	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+6 is mapped to this received priority
23	RESERVED	R	0h	
22-20	PRI5	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+5 is mapped to this received priority
19	RESERVED	R	0h	
18-16	PRI4	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+4 is mapped to this received priority
15	RESERVED	R	0h	
14-12	PRI3	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+3 is mapped to this received priority
11	RESERVED	R	0h	
10-8	PRI2	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+2 is mapped to this received priority
7	RESERVED	R	0h	
6-4	PRI1	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+1 is mapped to this received priority
3	RESERVED	R	0h	
2-0	PRI0	R/W	0h	A DSCP IPV4/V6 packet TOS of N*8+0 is mapped to this received priority

### 9.6.141 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_0 Register (Offset = 00022140h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_0 is shown in [Table 9-173](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-173. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.142 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_1 Register (Offset = 00022144h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_1 is shown in [Table 9-174](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-174. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.143 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_2 Register (Offset = 00022148h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_2 is shown in [Table 9-175](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-175. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.144 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_3 Register (Offset = 0002214Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_3 is shown in [Table 9-176](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-176. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.145 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_4 Register (Offset = 00022150h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_4 is shown in [Table 9-177](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-177. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.146 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_5 Register (Offset = 00022154h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_5 is shown in [Table 9-178](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-178. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate



### 9.6.147 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_6 Register (Offset = 00022158h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_6 is shown in [Table 9-179](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-179. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.148 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_7 Register (Offset = 0002215Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_7 is shown in [Table 9-180](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Committed Information Rate Value

**Table 9-180. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_CIR\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_CIR	R/W	0h	Priority N committed information rate

### 9.6.149 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_0 Register (Offset = 00022160h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_0 is shown in [Table 9-181](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-181. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.150 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_1 Register (Offset = 00022164h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_1 is shown in [Table 9-182](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-182. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.151 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_2 Register (Offset = 00022168h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_2 is shown in [Table 9-183](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-183. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.152 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_3 Register (Offset = 0002216Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_3 is shown in [Table 9-184](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-184. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.153 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_4 Register (Offset = 00022170h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_4 is shown in [Table 9-185](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-185. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.154 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_5 Register (Offset = 00022174h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_5 is shown in [Table 9-186](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-186. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count



### 9.6.155 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_6 Register (Offset = 00022178h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_6 is shown in [Table 9-187](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-187. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.156 CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_7 Register (Offset = 0002217Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_7 is shown in [Table 9-188](#).

Return to the [Summary Table](#).

Enet Port N Rx Priority P Excess Informatoin Rate Value

**Table 9-188. CPSW\_NC\_ETH\_MAC\_0\_PN\_PRI\_EIR\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R	0h	
27-0	PRI_EIR	R/W	0h	Priority N Excess Information Rate count

### 9.6.157 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_SET\_L\_REG Register (Offset = 00022180h) [Reset = 1FXFXFXFh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_SET\_L\_REG is shown in [Table 9-189](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Destination Threshold Set Low

**Table 9-189. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_SET\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 0

### 9.6.158 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_SET\_H\_REG Register (Offset = 00022184h) [Reset = 1FXFXFXFh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_SET\_H\_REG is shown in [Table 9-190](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Destination Threshold Set High

**Table 9-190. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_SET\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 4

### 9.6.159 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_CLR\_L\_REG Register (Offset = 00022188h) [Reset = 00X0X0X0h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_CLR\_L\_REG is shown in [Table 9-191](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Destination Threshold Clr Low

**Table 9-191. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_CLR\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 0

### 9.6.160 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_CLR\_H\_REG Register (Offset = 0002218Ch) [Reset = 00X0X0X0h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_CLR\_H\_REG is shown in [Table 9-192](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Destination Threshold Clr High

**Table 9-192. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_THRESH\_CLR\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 4

### 9.6.161 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_SET\_L\_REG Register (Offset = 00022190h) [Reset = 1FXFXFXFh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_SET\_L\_REG is shown in [Table 9-193](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Global Buffer Threshold Set Low

**Table 9-193. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_SET\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 0

### 9.6.162 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_SET\_H\_REG Register (Offset = 00022194h) [Reset = 1FXFXFXFh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_SET\_H\_REG is shown in [Table 9-194](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Global Buffer Threshold Set High

**Table 9-194. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_SET\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	1Fh	Port Priority Based Flow Control Threshold Set Value for Priority 4



### 9.6.163 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG Register (Offset = 00022198h) [Reset = 00X0X0X0h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG is shown in [Table 9-195](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Global Buffer Threshold Clr Low

**Table 9-195. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_CLR\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 0

### 9.6.164 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG Register (Offset = 0002219Ch) [Reset = 00X0X0X0h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG is shown in [Table 9-196](#).

Return to the [Summary Table](#).

Enet Port N Tx PFC Global Buffer Threshold Clr High

**Table 9-196. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_G\_BUF\_THRESH\_CLR\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	0h	Port Priority Based Flow Control Threshold Clear Value for Priority 4

### 9.6.165 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_OFLOW\_ADDVAL\_L\_REG Register (Offset = 00022300h) [Reset = 00X0X0X0h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_OFLOW\_ADDVAL\_L\_REG is shown in [Table 9-197](#).

Return to the [Summary Table](#).

Enet Port N Tx Destination Out Flow Add Values Low

**Table 9-197. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_OFLOW\_ADDVAL\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI3	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 3
23-21	RESERVED	R	0h	
20-16	PRI2	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 2
15-13	RESERVED	R	0h	
12-8	PRI1	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 1
7-5	RESERVED	R	0h	
4-0	PRI0	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 0

### 9.6.166 CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_OFLOW\_ADDVAL\_H\_REG Register (Offset = 00022304h) [Reset = 00X0X0X0h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_OFLOW\_ADDVAL\_H\_REG is shown in [Table 9-198](#).

Return to the [Summary Table](#).

Enet Port N Tx Destination Out Flow Add Values High

**Table 9-198. CPSW\_NC\_ETH\_MAC\_0\_PN\_TX\_D\_OFLOW\_ADDVAL\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28-24	PRI7	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 7
23-21	RESERVED	R	0h	
20-16	PRI6	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 6
15-13	RESERVED	R	0h	
12-8	PRI5	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 5
7-5	RESERVED	R	0h	
4-0	PRI4	R/W	0h	Port PFC Destination Based Out Flow Add Value for Priority 4

### 9.6.167 CPSW\_NC\_ETH\_MAC\_0\_PN\_SA\_L\_REG Register (Offset = 00022308h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_SA\_L\_REG is shown in [Table 9-199](#).

Return to the [Summary Table](#).

Enet Port N Tx Pause Frame Source Address Low

**Table 9-199. CPSW\_NC\_ETH\_MAC\_0\_PN\_SA\_L\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	MACSRCADDR_7_0	R/W	0h	Source Address Lower 8 bits
7-0	MACSRCADDR_15_8	R/W	0h	Source Address bits 15:8

### 9.6.168 CPSW\_NC\_ETH\_MAC\_0\_PN\_SA\_H\_REG Register (Offset = 0002230Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_SA\_H\_REG is shown in [Table 9-200](#).

Return to the [Summary Table](#).

Enet Port N Tx Pause Frame Source Address High

**Table 9-200. CPSW\_NC\_ETH\_MAC\_0\_PN\_SA\_H\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	MACSRCADDR_23_16	R/W	0h	Source Address bits 23:16
23-16	MACSRCADDR_31_24	R/W	0h	Source Address bits 31:24
15-8	MACSRCADDR_39_32	R/W	0h	Source Address bits 39:32
7-0	MACSRCADDR_47_40	R/W	0h	Source Address bits 47:40

### 9.6.169 CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL\_REG Register (Offset = 00022310h) [Reset = 0000X000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL\_REG is shown in [Table 9-201](#).

Return to the [Summary Table](#).

Enet Port N Time Sync Control

**Table 9-201. CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TS_MSG_TYPE_EN	R/W	0h	Time Sync Message Type Enable
15-12	RESERVED	R	0h	
11	TS_TX_HOST_TS_EN	R/W	0h	Time Sync Transmit Host Time Stamp Enable
10	TS_TX_ANNEX_E_EN	R/W	0h	Time Synce Transmit Annex E Enable
9	TS_RX_ANNEX_E_EN	R/W	0h	Time Synce Receive Annex E Enable
8	TS_LTYPE2_EN	R/W	0h	Time Sync LTYPE 2 enable transmit and receive
7	TS_TX_ANNEX_D_EN	R/W	0h	Time Synce Transmit Annex D Enable
6	TS_TX_VLAN_LTYPE2_EN	R/W	0h	Time Sync Transmit VLAN LTYPE 2 enable
5	TS_TX_VLAN_LTYPE1_EN	R/W	0h	Time Sync Transmit VLAN LTYPE 1 enable
4	TS_TX_ANNEX_F_EN	R/W	0h	Time Synce Transmit Annex F Enable
3	TS_RX_ANNEX_D_EN	R/W	0h	Time Synce Receive Annex D Enable
2	TS_RX_VLAN_LTYPE2_EN	R/W	0h	Time Sync Receive VLAN LTYPE 2 enable
1	TS_RX_VLAN_LTYPE1_EN	R/W	0h	Time Sync Receive VLAN LTYPE 1 enable
0	TS_RX_ANNEX_F_EN	R/W	0h	Time Synce Receive Annex F Enable

### 9.6.170 CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_SEQ\_LTYPE\_REG Register (Offset = 00022314h) [Reset = 001E0000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_SEQ\_LTYPE\_REG is shown in [Table 9-202](#).

Return to the [Summary Table](#).

Enet Port N Time Sync LTYPE (and SEQ\_ID\_OFFSET)

**Table 9-202. CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_SEQ\_LTYPE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-16	TS_SEQ_ID_OFFSET	R/W	1Eh	Time Sync Sequence ID Offset
15-0	TS_LTYPE1	R/W	0h	Time Sync LTYPE1



### 9.6.171 CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_VLAN\_LTYPE\_REG Register (Offset = 00022318h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_VLAN\_LTYPE\_REG is shown in [Table 9-203](#).

Return to the [Summary Table](#).

Enet Port N Time Sync VLAN2 and VLAN2

**Table 9-203. CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_VLAN\_LTYPE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TS_VLAN_LTYPE2	R/W	0h	Time Sync VLAN LTYPE2
15-0	TS_VLAN_LTYPE1	R/W	0h	Time Sync VLAN LTYPE1

### 9.6.172 CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL\_LTYPE2\_REG Register (Offset = 0002231Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL\_LTYPE2\_REG is shown in [Table 9-204](#).

Return to the [Summary Table](#).

Enet Port N Time Sync Control and LTYPE 2

**Table 9-204. CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL\_LTYPE2\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	TS_UNI_EN	R/W	0h	Time Sync Unicast Enable
23	TS_TTL_NONZERO	R/W	0h	Time Sync Time to Live Non-zero Enable
22	TS_320	R/W	0h	Time Sync Destination IP Address 320 Enable
21	TS_319	R/W	0h	Time Sync Destination IP Address 319 Enable
20	TS_132	R/W	0h	Time Sync Destination IP Address 132 Enable
19	TS_131	R/W	0h	Time Sync Destination IP Address 131 Enable
18	TS_130	R/W	0h	Time Sync Destination IP Address 130 Enable
17	TS_129	R/W	0h	Time Sync Destination IP Address 129 Enable
16	TS_107	R/W	0h	Time Sync Destination IP Address 107 Enable
15-0	TS_LTYPE2	R/W	0h	Time Sync LTYPE2

### 9.6.173 CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL2\_REG Register (Offset = 00022320h) [Reset = 00040000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL2\_REG is shown in [Table 9-205](#).

Return to the [Summary Table](#).

Enet Port N Time Sync Control 2

**Table 9-205. CPSW\_NC\_ETH\_MAC\_0\_PN\_TS\_CTL2\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-16	TS_DOMAIN_OFFSET	R/W	4h	Time Sync Domain Offset
15-0	TS_MCAST_TYPE_EN	R/W	0h	Time Sync Multicast Destination Address Type Enable

### 9.6.174 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_CONTROL\_REG Register (Offset = 00022330h) [Reset = 0000XX00h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_CONTROL\_REG is shown in [Table 9-206](#).

Return to the [Summary Table](#).

Enet Port N Mac Control

**Table 9-206. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	RX_CMF_EN	R/W	0h	RX Copy MAC Control Frames Enable
23	RX_CSF_EN	R/W	0h	RX Copy Short Frames Enable
22	RX_CEF_EN	R/W	0h	RX Copy Error Frames Enable
21	TX_SHORT_GAP_LIM_EN	R/W	0h	Transmit Short Gap Limit Enable
20	EXT_TX_FLOW_EN	R/W	0h	External Transmit Flow Control Enable
19	EXT_RX_FLOW_EN	R/W	0h	External Receive Flow Control Enable
18	EXT_EN	R/W	0h	External Enable
17	GIG_FORCE	R/W	0h	Gigabit Mode Force
16	IFCTL_B	R/W	0h	Interface Control B
15	IFCTL_A	R/W	0h	Interface Control A
14-13	RESERVED	R	0h	
12	CRC_TYPE	R/W	0h	Port CRC Type
11	CMD_IDLE	R/W	0h	Command Idle
10	TX_SHORT_GAP_ENABLE	R/W	0h	Transmit Short Gap Enable
9-8	RESERVED	R	0h	
7	GIG	R/W	0h	Gigabit Mode
6	TX_PACE	R/W	0h	Transmit Pacing Enable
5	GMII_EN	R/W	0h	GMII Enable
4	TX_FLOW_EN	R/W	0h	Transmit Flow Control Enable
3	RX_FLOW_EN	R/W	0h	Receive Flow Control Enable
2	MTEST	R/W	0h	Manufacturing Test Mode
1	LOOPBACK	R/W	0h	Loop Back Mode
0	FULLDUPLEX	R/W	0h	Full Duplex mode

### 9.6.175 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_STATUS\_REG Register (Offset = 00022334h) [Reset = X00000XXh]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_STATUS\_REG is shown in [Table 9-207](#).

Return to the [Summary Table](#).

Enet Port N Mac Status

**Table 9-207. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	cpxmac_sl IDLE
30	E_IDLE	R	1h	Express cpxmac_sl IDLE
29	RESERVED	R	0h	
28	MAC_TX_IDLE	R	1h	Prempt and Express cpxmac_sl Transmit IDLE
27	TORF	R	0h	Top of receive FIFO flow control trigger occurred. This bit is write one to clear.
26-24	TORF_PRI	R	0h	The lowest priority that caused top of receive FIFO flow control trigger since the last write to clear. This field is write 0x7 to clear.
23-16	TX_PFC_FLOW_ACT	R	0h	Transmit Priority Based Flow Control Active (priority 7 down to 0)
15-8	RX_PFC_FLOW_ACT	R	0h	Receive Priority Based Flow Control Active (priority 7 down to 0)
7	RESERVED	R	0h	
6	EXT_RX_FLOW_EN	R	0h	External Transmit Flow Control Enable
5	EXT_TX_FLOW_EN	R	0h	External Receive Flow Control Enable
4	EXT_GIG	R	0h	External GIG mode
3	EXT_FULLDUPLEX	R	0h	External Fullduplex
2	RESERVED	R	0h	
1	RX_FLOW_ACT	R	0h	Receive Flow Control Active
0	TX_FLOW_ACT	R	0h	Transmit Flow Control Active

### 9.6.176 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_SOFT\_RESET\_REG Register (Offset = 00022338h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_SOFT\_RESET\_REG is shown in [Table 9-208](#).

Return to the [Summary Table](#).

Enet Port N Mac Soft Reset

**Table 9-208. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_SOFT\_RESET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_RESET	R/W	0h	Software reset

### 9.6.177 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_BOFFTEST\_REG Register (Offset = 0002233Ch) [Reset = 00000X00h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_BOFFTEST\_REG is shown in [Table 9-209](#).

Return to the [Summary Table](#).

Enet Port N Mac Backoff Test

**Table 9-209. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_BOFFTEST\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RESERVED	R	0h	
30-26	PACEVAL	R/W	0h	Pacing Register Current Value
25-16	RNDNUM	R/W	0h	Backoff Random Number Generator
15-12	COLL_COUNT	R	0h	Collision Count
11-10	RESERVED	R	0h	
9-0	TX_BACKOFF	R	0h	Backoff Count

### 9.6.178 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RX\_PAUSETIMER\_REG Register (Offset = 00022340h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RX\_PAUSETIMER\_REG is shown in [Table 9-210](#).

Return to the [Summary Table](#).

Enet Port N 802.3 Receive Pause Timer

**Table 9-210. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RX\_PAUSETIMER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value



### 9.6.179 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_0 Register (Offset = 00022350h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_0 is shown in [Table 9-211](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-211. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.180 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_1 Register (Offset = 00022354h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_1 is shown in [Table 9-212](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-212. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.181 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_2 Register (Offset = 00022358h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_2 is shown in [Table 9-213](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-213. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.182 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_3 Register (Offset = 0002235Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_3 is shown in [Table 9-214](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-214. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.183 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_4 Register (Offset = 00022360h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_4 is shown in [Table 9-215](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-215. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.184 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_5 Register (Offset = 00022364h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_5 is shown in [Table 9-216](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-216. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.185 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_6 Register (Offset = 00022368h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_6 is shown in [Table 9-217](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-217. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value

### 9.6.186 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_7 Register (Offset = 0002236Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_7 is shown in [Table 9-218](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Rx Pause Timer

**Table 9-218. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_RXN\_PAUSETIMER\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	RX_PAUSETIMER	R/W	0h	RX Pause Timer Value



### 9.6.187 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TX\_PAUSETIMER\_REG Register (Offset = 00022370h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TX\_PAUSETIMER\_REG is shown in [Table 9-219](#).

Return to the [Summary Table](#).

Enet Port N 802.3 Tx Pause Timer

**Table 9-219. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TX\_PAUSETIMER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.188 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_0 Register (Offset = 00022380h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_0 is shown in [Table 9-220](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-220. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.189 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_1 Register (Offset = 00022384h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_1 is shown in [Table 9-221](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-221. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.190 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_2 Register (Offset = 00022388h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_2 is shown in [Table 9-222](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-222. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.191 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_3 Register (Offset = 0002238Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_3 is shown in [Table 9-223](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-223. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.192 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_4 Register (Offset = 00022390h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_4 is shown in [Table 9-224](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-224. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.193 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_5 Register (Offset = 00022394h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_5 is shown in [Table 9-225](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-225. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.194 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_6 Register (Offset = 00022398h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_6 is shown in [Table 9-226](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-226. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value



### 9.6.195 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_7 Register (Offset = 0002239Ch) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_7 is shown in [Table 9-227](#).

Return to the [Summary Table](#).

Enet Port N PFC Priority P Tx Pause Timer

**Table 9-227. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TXN\_PAUSETIMER\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_PAUSETIMER	R/W	0h	TX Pause Timer Value

### 9.6.196 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_EMCONTROL\_REG Register (Offset = 000223A0h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_EMCONTROL\_REG is shown in [Table 9-228](#).

Return to the [Summary Table](#).

Enet Port N Emulation Control

**Table 9-228. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_EMCONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	SOFT	R/W	0h	Emulation Soft Bit
0	FREE	R/W	0h	Emulation Free Bit

### 9.6.197 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TX\_GAP\_REG Register (Offset = 000223A4h) [Reset = 0000000Ch]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TX\_GAP\_REG is shown in [Table 9-229](#).

Return to the [Summary Table](#).

Enet Port N Tx Inter Packet Gap

**Table 9-229. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_TX\_GAP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	TX_GAP	R/W	Ch	Transmit Inter-Packet Gap

### 9.6.198 CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_PORT\_CONFIG Register (Offset = 000223A8h) [Reset = 00000004h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_PORT\_CONFIG is shown in [Table 9-230](#).

Return to the [Summary Table](#).

Enet Port N Port Configuration

**Table 9-230. CPSW\_NC\_ETH\_MAC\_0\_PN\_MAC\_PORT\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9	IET	R	0h	IET support
8	XGMII	R	0h	No XGMII support
7-0	INTERVLAN_ROUTES	R	4h	The number of InterVLAN routes

### 9.6.199 CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_POINTER\_REG Register (Offset = 000223ACh) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_POINTER\_REG is shown in [Table 9-231](#).

Return to the [Summary Table](#).

Enet Port N Tx Egress InterVLAN Operation Pointer

**Table 9-231. CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_POINTER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	POINTER	R/W	0h	InterVLAN location pointer: This field points to the InterVLAN location that will be read/written by accesses to Enet_Pn_InterVLANx_A/B.

### 9.6.200 CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_A\_REG Register (Offset = 000223B0h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_A\_REG is shown in [Table 9-232](#).

Return to the [Summary Table](#).

Enet Port N Tx Egress InterVLAN A

**Table 9-232. CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_A\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	DA_23_16	R/W	0h	Destination Address bits 23:16
23-16	DA_31_24	R/W	0h	Destination Address bits 31:24
15-8	DA_39_32	R/W	0h	Destination Address bits 39:32
7-0	DA_47_40	R/W	0h	Destination Address bits 47:40

### 9.6.201 CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_B\_REG Register (Offset = 000223B4h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_B\_REG is shown in [Table 9-233](#).

Return to the [Summary Table](#).

Enet Port N Tx Egress InterVLAN B

**Table 9-233. CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_B\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	SA_39_32	R/W	0h	Source Address bits 39:32
23-16	SA_47_40	R/W	0h	Source Address bits 47:40
15-8	DA_7_0	R/W	0h	Destination Address bits 7:0
7-0	DA_15_8	R/W	0h	Destination Address bits 15:8

### 9.6.202 CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_C\_REG Register (Offset = 000223B8h) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_C\_REG is shown in [Table 9-234](#).

Return to the [Summary Table](#).

Enet Port N Tx Egress InterVLAN C

**Table 9-234. CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_C\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	SA_7_0	R/W	0h	Source Address bits 7:0
23-16	SA_15_8	R/W	0h	Source Address bits 15:8
15-8	SA_23_16	R/W	0h	Source Address bits 23:16
7-0	SA_31_24	R/W	0h	Source Address bits 31:24



### 9.6.203 CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_D\_REG Register (Offset = 000223BCh) [Reset = 00000000h]

CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_D\_REG is shown in [Table 9-235](#).

Return to the [Summary Table](#).

Enet Port N Tx Egress InterVLAN D

**Table 9-235. CPSW\_NC\_ETH\_MAC\_0\_PN\_INTERVLAN\_OPX\_D\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DECREMENT_TTL	R/W	0h	Decrement Time To Live: When set, the Time To Live (TTL) field in the header is decremented.
14	DEST_FORCE_UNTAGGED_EGRESS	R/W	0h	Destination VLAN Force Untagged Egress: When set, this bit indicates that the VLAN should be removed on egress for the routed packet.
13	REPLACE_DA_SA	R/W	0h	Replace Destination Address and Source Address: When set this bit indicates that the routed packet destination address should be replaced by da[47:0] and the source address should be replaced by sa[47:0].
12	REPLACE_VID	R/W	0h	Replace VLAN ID: When set this bit indicates that the VLAN ID should be replaced for the routed packet.
11-0	VID	R/W	0h	VLAN ID

### 9.6.204 CPSW\_NC\_EST\_FETCH\_LOC\_0 Register (Offset = 00032000h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_0 is shown in [Table 9-236](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-236. CPSW\_NC\_EST\_FETCH\_LOC\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.205 CPSW\_NC\_EST\_FETCH\_LOC\_1 Register (Offset = 00032004h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_1 is shown in [Table 9-237](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-237. CPSW\_NC\_EST\_FETCH\_LOC\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.206 CPSW\_NC\_EST\_FETCH\_LOC\_2 Register (Offset = 00032008h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_2 is shown in [Table 9-238](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-238. CPSW\_NC\_EST\_FETCH\_LOC\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.207 CPSW\_NC\_EST\_FETCH\_LOC\_3 Register (Offset = 0003200Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_3 is shown in [Table 9-239](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-239. CPSW\_NC\_EST\_FETCH\_LOC\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.208 CPSW\_NC\_EST\_FETCH\_LOC\_4 Register (Offset = 00032010h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_4 is shown in [Table 9-240](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-240. CPSW\_NC\_EST\_FETCH\_LOC\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.209 CPSW\_NC\_EST\_FETCH\_LOC\_5 Register (Offset = 00032014h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_5 is shown in [Table 9-241](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-241. CPSW\_NC\_EST\_FETCH\_LOC\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.210 CPSW\_NC\_EST\_FETCH\_LOC\_6 Register (Offset = 00032018h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_6 is shown in [Table 9-242](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-242. CPSW\_NC\_EST\_FETCH\_LOC\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.211 CPSW\_NC\_EST\_FETCH\_LOC\_7 Register (Offset = 0003201Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_7 is shown in [Table 9-243](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-243. CPSW\_NC\_EST\_FETCH\_LOC\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.212 CPSW\_NC\_EST\_FETCH\_LOC\_8 Register (Offset = 00032020h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_8 is shown in [Table 9-244](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-244. CPSW\_NC\_EST\_FETCH\_LOC\_8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.213 CPSW\_NC\_EST\_FETCH\_LOC\_9 Register (Offset = 00032024h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_9 is shown in [Table 9-245](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-245. CPSW\_NC\_EST\_FETCH\_LOC\_9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.214 CPSW\_NC\_EST\_FETCH\_LOC\_10 Register (Offset = 00032028h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_10 is shown in [Table 9-246](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-246. CPSW\_NC\_EST\_FETCH\_LOC\_10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.215 CPSW\_NC\_EST\_FETCH\_LOC\_11 Register (Offset = 0003202Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_11 is shown in [Table 9-247](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-247. CPSW\_NC\_EST\_FETCH\_LOC\_11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.216 CPSW\_NC\_EST\_FETCH\_LOC\_12 Register (Offset = 00032030h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_12 is shown in [Table 9-248](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-248. CPSW\_NC\_EST\_FETCH\_LOC\_12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.217 CPSW\_NC\_EST\_FETCH\_LOC\_13 Register (Offset = 00032034h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_13 is shown in [Table 9-249](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-249. CPSW\_NC\_EST\_FETCH\_LOC\_13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.218 CPSW\_NC\_EST\_FETCH\_LOC\_14 Register (Offset = 00032038h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_14 is shown in [Table 9-250](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-250. CPSW\_NC\_EST\_FETCH\_LOC\_14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.219 CPSW\_NC\_EST\_FETCH\_LOC\_15 Register (Offset = 0003203Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_15 is shown in [Table 9-251](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-251. CPSW\_NC\_EST\_FETCH\_LOC\_15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.220 CPSW\_NC\_EST\_FETCH\_LOC\_16 Register (Offset = 00032040h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_16 is shown in [Table 9-252](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-252. CPSW\_NC\_EST\_FETCH\_LOC\_16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.221 CPSW\_NC\_EST\_FETCH\_LOC\_17 Register (Offset = 00032044h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_17 is shown in [Table 9-253](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-253. CPSW\_NC\_EST\_FETCH\_LOC\_17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.222 CPSW\_NC\_EST\_FETCH\_LOC\_18 Register (Offset = 00032048h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_18 is shown in [Table 9-254](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-254. CPSW\_NC\_EST\_FETCH\_LOC\_18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.223 CPSW\_NC\_EST\_FETCH\_LOC\_19 Register (Offset = 0003204Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_19 is shown in [Table 9-255](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-255. CPSW\_NC\_EST\_FETCH\_LOC\_19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.224 CPSW\_NC\_EST\_FETCH\_LOC\_20 Register (Offset = 00032050h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_20 is shown in [Table 9-256](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-256. CPSW\_NC\_EST\_FETCH\_LOC\_20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.225 CPSW\_NC\_EST\_FETCH\_LOC\_21 Register (Offset = 00032054h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_21 is shown in [Table 9-257](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-257. CPSW\_NC\_EST\_FETCH\_LOC\_21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.226 CPSW\_NC\_EST\_FETCH\_LOC\_22 Register (Offset = 00032058h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_22 is shown in [Table 9-258](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-258. CPSW\_NC\_EST\_FETCH\_LOC\_22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.227 CPSW\_NC\_EST\_FETCH\_LOC\_23 Register (Offset = 0003205Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_23 is shown in [Table 9-259](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-259. CPSW\_NC\_EST\_FETCH\_LOC\_23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.228 CPSW\_NC\_EST\_FETCH\_LOC\_24 Register (Offset = 00032060h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_24 is shown in [Table 9-260](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-260. CPSW\_NC\_EST\_FETCH\_LOC\_24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.229 CPSW\_NC\_EST\_FETCH\_LOC\_25 Register (Offset = 00032064h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_25 is shown in [Table 9-261](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-261. CPSW\_NC\_EST\_FETCH\_LOC\_25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.230 CPSW\_NC\_EST\_FETCH\_LOC\_26 Register (Offset = 00032068h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_26 is shown in [Table 9-262](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-262. CPSW\_NC\_EST\_FETCH\_LOC\_26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.231 CPSW\_NC\_EST\_FETCH\_LOC\_27 Register (Offset = 0003206Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_27 is shown in [Table 9-263](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-263. CPSW\_NC\_EST\_FETCH\_LOC\_27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.232 CPSW\_NC\_EST\_FETCH\_LOC\_28 Register (Offset = 00032070h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_28 is shown in [Table 9-264](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-264. CPSW\_NC\_EST\_FETCH\_LOC\_28 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.233 CPSW\_NC\_EST\_FETCH\_LOC\_29 Register (Offset = 00032074h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_29 is shown in [Table 9-265](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-265. CPSW\_NC\_EST\_FETCH\_LOC\_29 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.234 CPSW\_NC\_EST\_FETCH\_LOC\_30 Register (Offset = 00032078h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_30 is shown in [Table 9-266](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-266. CPSW\_NC\_EST\_FETCH\_LOC\_30 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.235 CPSW\_NC\_EST\_FETCH\_LOC\_31 Register (Offset = 0003207Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_31 is shown in [Table 9-267](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-267. CPSW\_NC\_EST\_FETCH\_LOC\_31 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.236 CPSW\_NC\_EST\_FETCH\_LOC\_32 Register (Offset = 00032080h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_32 is shown in [Table 9-268](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-268. CPSW\_NC\_EST\_FETCH\_LOC\_32 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.237 CPSW\_NC\_EST\_FETCH\_LOC\_33 Register (Offset = 00032084h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_33 is shown in [Table 9-269](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-269. CPSW\_NC\_EST\_FETCH\_LOC\_33 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.238 CPSW\_NC\_EST\_FETCH\_LOC\_34 Register (Offset = 00032088h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_34 is shown in [Table 9-270](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-270. CPSW\_NC\_EST\_FETCH\_LOC\_34 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.239 CPSW\_NC\_EST\_FETCH\_LOC\_35 Register (Offset = 0003208Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_35 is shown in [Table 9-271](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-271. CPSW\_NC\_EST\_FETCH\_LOC\_35 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.240 CPSW\_NC\_EST\_FETCH\_LOC\_36 Register (Offset = 00032090h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_36 is shown in [Table 9-272](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-272. CPSW\_NC\_EST\_FETCH\_LOC\_36 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.241 CPSW\_NC\_EST\_FETCH\_LOC\_37 Register (Offset = 00032094h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_37 is shown in [Table 9-273](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-273. CPSW\_NC\_EST\_FETCH\_LOC\_37 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.242 CPSW\_NC\_EST\_FETCH\_LOC\_38 Register (Offset = 00032098h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_38 is shown in [Table 9-274](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-274. CPSW\_NC\_EST\_FETCH\_LOC\_38 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.243 CPSW\_NC\_EST\_FETCH\_LOC\_39 Register (Offset = 0003209Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_39 is shown in [Table 9-275](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-275. CPSW\_NC\_EST\_FETCH\_LOC\_39 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.244 CPSW\_NC\_EST\_FETCH\_LOC\_40 Register (Offset = 000320A0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_40 is shown in [Table 9-276](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-276. CPSW\_NC\_EST\_FETCH\_LOC\_40 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.245 CPSW\_NC\_EST\_FETCH\_LOC\_41 Register (Offset = 000320A4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_41 is shown in [Table 9-277](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-277. CPSW\_NC\_EST\_FETCH\_LOC\_41 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.246 CPSW\_NC\_EST\_FETCH\_LOC\_42 Register (Offset = 000320A8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_42 is shown in [Table 9-278](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-278. CPSW\_NC\_EST\_FETCH\_LOC\_42 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.247 CPSW\_NC\_EST\_FETCH\_LOC\_43 Register (Offset = 000320ACh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_43 is shown in [Table 9-279](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-279. CPSW\_NC\_EST\_FETCH\_LOC\_43 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.248 CPSW\_NC\_EST\_FETCH\_LOC\_44 Register (Offset = 000320B0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_44 is shown in [Table 9-280](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-280. CPSW\_NC\_EST\_FETCH\_LOC\_44 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.249 CPSW\_NC\_EST\_FETCH\_LOC\_45 Register (Offset = 000320B4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_45 is shown in [Table 9-281](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-281. CPSW\_NC\_EST\_FETCH\_LOC\_45 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.250 CPSW\_NC\_EST\_FETCH\_LOC\_46 Register (Offset = 000320B8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_46 is shown in [Table 9-282](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-282. CPSW\_NC\_EST\_FETCH\_LOC\_46 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.251 CPSW\_NC\_EST\_FETCH\_LOC\_47 Register (Offset = 000320BCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_47 is shown in [Table 9-283](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-283. CPSW\_NC\_EST\_FETCH\_LOC\_47 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.252 CPSW\_NC\_EST\_FETCH\_LOC\_48 Register (Offset = 000320C0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_48 is shown in [Table 9-284](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-284. CPSW\_NC\_EST\_FETCH\_LOC\_48 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.253 CPSW\_NC\_EST\_FETCH\_LOC\_49 Register (Offset = 000320C4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_49 is shown in [Table 9-285](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-285. CPSW\_NC\_EST\_FETCH\_LOC\_49 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.254 CPSW\_NC\_EST\_FETCH\_LOC\_50 Register (Offset = 000320C8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_50 is shown in [Table 9-286](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-286. CPSW\_NC\_EST\_FETCH\_LOC\_50 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.255 CPSW\_NC\_EST\_FETCH\_LOC\_51 Register (Offset = 000320CCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_51 is shown in [Table 9-287](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-287. CPSW\_NC\_EST\_FETCH\_LOC\_51 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.256 CPSW\_NC\_EST\_FETCH\_LOC\_52 Register (Offset = 000320D0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_52 is shown in [Table 9-288](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-288. CPSW\_NC\_EST\_FETCH\_LOC\_52 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.257 CPSW\_NC\_EST\_FETCH\_LOC\_53 Register (Offset = 000320D4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_53 is shown in [Table 9-289](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-289. CPSW\_NC\_EST\_FETCH\_LOC\_53 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.258 CPSW\_NC\_EST\_FETCH\_LOC\_54 Register (Offset = 000320D8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_54 is shown in [Table 9-290](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-290. CPSW\_NC\_EST\_FETCH\_LOC\_54 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.259 CPSW\_NC\_EST\_FETCH\_LOC\_55 Register (Offset = 000320DCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_55 is shown in [Table 9-291](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-291. CPSW\_NC\_EST\_FETCH\_LOC\_55 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.260 CPSW\_NC\_EST\_FETCH\_LOC\_56 Register (Offset = 000320E0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_56 is shown in [Table 9-292](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-292. CPSW\_NC\_EST\_FETCH\_LOC\_56 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.261 CPSW\_NC\_EST\_FETCH\_LOC\_57 Register (Offset = 000320E4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_57 is shown in [Table 9-293](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-293. CPSW\_NC\_EST\_FETCH\_LOC\_57 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.262 CPSW\_NC\_EST\_FETCH\_LOC\_58 Register (Offset = 000320E8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_58 is shown in [Table 9-294](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-294. CPSW\_NC\_EST\_FETCH\_LOC\_58 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.263 CPSW\_NC\_EST\_FETCH\_LOC\_59 Register (Offset = 000320ECh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_59 is shown in [Table 9-295](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-295. CPSW\_NC\_EST\_FETCH\_LOC\_59 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.264 CPSW\_NC\_EST\_FETCH\_LOC\_60 Register (Offset = 000320F0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_60 is shown in [Table 9-296](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-296. CPSW\_NC\_EST\_FETCH\_LOC\_60 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.265 CPSW\_NC\_EST\_FETCH\_LOC\_61 Register (Offset = 000320F4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_61 is shown in [Table 9-297](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-297. CPSW\_NC\_EST\_FETCH\_LOC\_61 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.266 CPSW\_NC\_EST\_FETCH\_LOC\_62 Register (Offset = 000320F8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_62 is shown in [Table 9-298](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-298. CPSW\_NC\_EST\_FETCH\_LOC\_62 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.267 CPSW\_NC\_EST\_FETCH\_LOC\_63 Register (Offset = 000320FCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_63 is shown in [Table 9-299](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-299. CPSW\_NC\_EST\_FETCH\_LOC\_63 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.268 CPSW\_NC\_EST\_FETCH\_LOC\_64 Register (Offset = 00032100h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_64 is shown in [Table 9-300](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-300. CPSW\_NC\_EST\_FETCH\_LOC\_64 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.269 CPSW\_NC\_EST\_FETCH\_LOC\_65 Register (Offset = 00032104h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_65 is shown in [Table 9-301](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-301. CPSW\_NC\_EST\_FETCH\_LOC\_65 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.270 CPSW\_NC\_EST\_FETCH\_LOC\_66 Register (Offset = 00032108h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_66 is shown in [Table 9-302](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-302. CPSW\_NC\_EST\_FETCH\_LOC\_66 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.271 CPSW\_NC\_EST\_FETCH\_LOC\_67 Register (Offset = 0003210Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_67 is shown in [Table 9-303](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-303. CPSW\_NC\_EST\_FETCH\_LOC\_67 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.272 CPSW\_NC\_EST\_FETCH\_LOC\_68 Register (Offset = 00032110h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_68 is shown in [Table 9-304](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-304. CPSW\_NC\_EST\_FETCH\_LOC\_68 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.273 CPSW\_NC\_EST\_FETCH\_LOC\_69 Register (Offset = 00032114h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_69 is shown in [Table 9-305](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-305. CPSW\_NC\_EST\_FETCH\_LOC\_69 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.274 CPSW\_NC\_EST\_FETCH\_LOC\_70 Register (Offset = 00032118h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_70 is shown in [Table 9-306](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-306. CPSW\_NC\_EST\_FETCH\_LOC\_70 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.275 CPSW\_NC\_EST\_FETCH\_LOC\_71 Register (Offset = 0003211Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_71 is shown in [Table 9-307](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-307. CPSW\_NC\_EST\_FETCH\_LOC\_71 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.276 CPSW\_NC\_EST\_FETCH\_LOC\_72 Register (Offset = 00032120h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_72 is shown in [Table 9-308](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-308. CPSW\_NC\_EST\_FETCH\_LOC\_72 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.277 CPSW\_NC\_EST\_FETCH\_LOC\_73 Register (Offset = 00032124h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_73 is shown in [Table 9-309](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-309. CPSW\_NC\_EST\_FETCH\_LOC\_73 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.278 CPSW\_NC\_EST\_FETCH\_LOC\_74 Register (Offset = 00032128h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_74 is shown in [Table 9-310](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-310. CPSW\_NC\_EST\_FETCH\_LOC\_74 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.279 CPSW\_NC\_EST\_FETCH\_LOC\_75 Register (Offset = 0003212Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_75 is shown in [Table 9-311](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-311. CPSW\_NC\_EST\_FETCH\_LOC\_75 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.280 CPSW\_NC\_EST\_FETCH\_LOC\_76 Register (Offset = 00032130h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_76 is shown in [Table 9-312](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-312. CPSW\_NC\_EST\_FETCH\_LOC\_76 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.281 CPSW\_NC\_EST\_FETCH\_LOC\_77 Register (Offset = 00032134h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_77 is shown in [Table 9-313](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-313. CPSW\_NC\_EST\_FETCH\_LOC\_77 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.282 CPSW\_NC\_EST\_FETCH\_LOC\_78 Register (Offset = 00032138h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_78 is shown in [Table 9-314](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-314. CPSW\_NC\_EST\_FETCH\_LOC\_78 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.283 CPSW\_NC\_EST\_FETCH\_LOC\_79 Register (Offset = 0003213Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_79 is shown in [Table 9-315](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-315. CPSW\_NC\_EST\_FETCH\_LOC\_79 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.284 CPSW\_NC\_EST\_FETCH\_LOC\_80 Register (Offset = 00032140h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_80 is shown in [Table 9-316](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-316. CPSW\_NC\_EST\_FETCH\_LOC\_80 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.285 CPSW\_NC\_EST\_FETCH\_LOC\_81 Register (Offset = 00032144h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_81 is shown in [Table 9-317](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-317. CPSW\_NC\_EST\_FETCH\_LOC\_81 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.286 CPSW\_NC\_EST\_FETCH\_LOC\_82 Register (Offset = 00032148h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_82 is shown in [Table 9-318](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-318. CPSW\_NC\_EST\_FETCH\_LOC\_82 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.287 CPSW\_NC\_EST\_FETCH\_LOC\_83 Register (Offset = 0003214Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_83 is shown in [Table 9-319](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-319. CPSW\_NC\_EST\_FETCH\_LOC\_83 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.288 CPSW\_NC\_EST\_FETCH\_LOC\_84 Register (Offset = 00032150h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_84 is shown in [Table 9-320](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-320. CPSW\_NC\_EST\_FETCH\_LOC\_84 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.289 CPSW\_NC\_EST\_FETCH\_LOC\_85 Register (Offset = 00032154h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_85 is shown in [Table 9-321](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-321. CPSW\_NC\_EST\_FETCH\_LOC\_85 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.290 CPSW\_NC\_EST\_FETCH\_LOC\_86 Register (Offset = 00032158h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_86 is shown in [Table 9-322](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-322. CPSW\_NC\_EST\_FETCH\_LOC\_86 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.291 CPSW\_NC\_EST\_FETCH\_LOC\_87 Register (Offset = 0003215Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_87 is shown in [Table 9-323](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-323. CPSW\_NC\_EST\_FETCH\_LOC\_87 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.292 CPSW\_NC\_EST\_FETCH\_LOC\_88 Register (Offset = 00032160h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_88 is shown in [Table 9-324](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-324. CPSW\_NC\_EST\_FETCH\_LOC\_88 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.293 CPSW\_NC\_EST\_FETCH\_LOC\_89 Register (Offset = 00032164h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_89 is shown in [Table 9-325](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-325. CPSW\_NC\_EST\_FETCH\_LOC\_89 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.294 CPSW\_NC\_EST\_FETCH\_LOC\_90 Register (Offset = 00032168h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_90 is shown in [Table 9-326](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-326. CPSW\_NC\_EST\_FETCH\_LOC\_90 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.295 CPSW\_NC\_EST\_FETCH\_LOC\_91 Register (Offset = 0003216Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_91 is shown in [Table 9-327](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-327. CPSW\_NC\_EST\_FETCH\_LOC\_91 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.296 CPSW\_NC\_EST\_FETCH\_LOC\_92 Register (Offset = 00032170h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_92 is shown in [Table 9-328](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-328. CPSW\_NC\_EST\_FETCH\_LOC\_92 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.297 CPSW\_NC\_EST\_FETCH\_LOC\_93 Register (Offset = 00032174h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_93 is shown in [Table 9-329](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-329. CPSW\_NC\_EST\_FETCH\_LOC\_93 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.298 CPSW\_NC\_EST\_FETCH\_LOC\_94 Register (Offset = 00032178h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_94 is shown in [Table 9-330](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-330. CPSW\_NC\_EST\_FETCH\_LOC\_94 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.299 CPSW\_NC\_EST\_FETCH\_LOC\_95 Register (Offset = 0003217Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_95 is shown in [Table 9-331](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-331. CPSW\_NC\_EST\_FETCH\_LOC\_95 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.300 CPSW\_NC\_EST\_FETCH\_LOC\_96 Register (Offset = 00032180h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_96 is shown in [Table 9-332](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-332. CPSW\_NC\_EST\_FETCH\_LOC\_96 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.301 CPSW\_NC\_EST\_FETCH\_LOC\_97 Register (Offset = 00032184h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_97 is shown in [Table 9-333](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-333. CPSW\_NC\_EST\_FETCH\_LOC\_97 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.302 CPSW\_NC\_EST\_FETCH\_LOC\_98 Register (Offset = 00032188h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_98 is shown in [Table 9-334](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-334. CPSW\_NC\_EST\_FETCH\_LOC\_98 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.303 CPSW\_NC\_EST\_FETCH\_LOC\_99 Register (Offset = 0003218Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_99 is shown in [Table 9-335](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-335. CPSW\_NC\_EST\_FETCH\_LOC\_99 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.304 CPSW\_NC\_EST\_FETCH\_LOC\_100 Register (Offset = 00032190h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_100 is shown in [Table 9-336](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-336. CPSW\_NC\_EST\_FETCH\_LOC\_100 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.305 CPSW\_NC\_EST\_FETCH\_LOC\_101 Register (Offset = 00032194h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_101 is shown in [Table 9-337](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-337. CPSW\_NC\_EST\_FETCH\_LOC\_101 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.306 CPSW\_NC\_EST\_FETCH\_LOC\_102 Register (Offset = 00032198h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_102 is shown in [Table 9-338](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-338. CPSW\_NC\_EST\_FETCH\_LOC\_102 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.307 CPSW\_NC\_EST\_FETCH\_LOC\_103 Register (Offset = 0003219Ch) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_103 is shown in [Table 9-339](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-339. CPSW\_NC\_EST\_FETCH\_LOC\_103 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.308 CPSW\_NC\_EST\_FETCH\_LOC\_104 Register (Offset = 000321A0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_104 is shown in [Table 9-340](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-340. CPSW\_NC\_EST\_FETCH\_LOC\_104 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.309 CPSW\_NC\_EST\_FETCH\_LOC\_105 Register (Offset = 000321A4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_105 is shown in [Table 9-341](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-341. CPSW\_NC\_EST\_FETCH\_LOC\_105 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.310 CPSW\_NC\_EST\_FETCH\_LOC\_106 Register (Offset = 000321A8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_106 is shown in [Table 9-342](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-342. CPSW\_NC\_EST\_FETCH\_LOC\_106 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.311 CPSW\_NC\_EST\_FETCH\_LOC\_107 Register (Offset = 000321ACh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_107 is shown in [Table 9-343](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-343. CPSW\_NC\_EST\_FETCH\_LOC\_107 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.312 CPSW\_NC\_EST\_FETCH\_LOC\_108 Register (Offset = 000321B0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_108 is shown in [Table 9-344](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-344. CPSW\_NC\_EST\_FETCH\_LOC\_108 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.313 CPSW\_NC\_EST\_FETCH\_LOC\_109 Register (Offset = 000321B4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_109 is shown in [Table 9-345](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-345. CPSW\_NC\_EST\_FETCH\_LOC\_109 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.314 CPSW\_NC\_EST\_FETCH\_LOC\_110 Register (Offset = 000321B8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_110 is shown in [Table 9-346](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-346. CPSW\_NC\_EST\_FETCH\_LOC\_110 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.315 CPSW\_NC\_EST\_FETCH\_LOC\_111 Register (Offset = 000321BCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_111 is shown in [Table 9-347](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-347. CPSW\_NC\_EST\_FETCH\_LOC\_111 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.316 CPSW\_NC\_EST\_FETCH\_LOC\_112 Register (Offset = 000321C0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_112 is shown in [Table 9-348](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-348. CPSW\_NC\_EST\_FETCH\_LOC\_112 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.317 CPSW\_NC\_EST\_FETCH\_LOC\_113 Register (Offset = 000321C4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_113 is shown in [Table 9-349](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-349. CPSW\_NC\_EST\_FETCH\_LOC\_113 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.318 CPSW\_NC\_EST\_FETCH\_LOC\_114 Register (Offset = 000321C8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_114 is shown in [Table 9-350](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-350. CPSW\_NC\_EST\_FETCH\_LOC\_114 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.319 CPSW\_NC\_EST\_FETCH\_LOC\_115 Register (Offset = 000321CCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_115 is shown in [Table 9-351](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-351. CPSW\_NC\_EST\_FETCH\_LOC\_115 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.320 CPSW\_NC\_EST\_FETCH\_LOC\_116 Register (Offset = 000321D0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_116 is shown in [Table 9-352](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-352. CPSW\_NC\_EST\_FETCH\_LOC\_116 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.321 CPSW\_NC\_EST\_FETCH\_LOC\_117 Register (Offset = 000321D4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_117 is shown in [Table 9-353](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-353. CPSW\_NC\_EST\_FETCH\_LOC\_117 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.322 CPSW\_NC\_EST\_FETCH\_LOC\_118 Register (Offset = 000321D8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_118 is shown in [Table 9-354](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-354. CPSW\_NC\_EST\_FETCH\_LOC\_118 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.323 CPSW\_NC\_EST\_FETCH\_LOC\_119 Register (Offset = 000321DCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_119 is shown in [Table 9-355](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-355. CPSW\_NC\_EST\_FETCH\_LOC\_119 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.324 CPSW\_NC\_EST\_FETCH\_LOC\_120 Register (Offset = 000321E0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_120 is shown in [Table 9-356](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-356. CPSW\_NC\_EST\_FETCH\_LOC\_120 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.325 CPSW\_NC\_EST\_FETCH\_LOC\_121 Register (Offset = 000321E4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_121 is shown in [Table 9-357](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-357. CPSW\_NC\_EST\_FETCH\_LOC\_121 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.326 CPSW\_NC\_EST\_FETCH\_LOC\_122 Register (Offset = 000321E8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_122 is shown in [Table 9-358](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-358. CPSW\_NC\_EST\_FETCH\_LOC\_122 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.327 CPSW\_NC\_EST\_FETCH\_LOC\_123 Register (Offset = 000321ECh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_123 is shown in [Table 9-359](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-359. CPSW\_NC\_EST\_FETCH\_LOC\_123 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.328 CPSW\_NC\_EST\_FETCH\_LOC\_124 Register (Offset = 000321F0h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_124 is shown in [Table 9-360](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-360. CPSW\_NC\_EST\_FETCH\_LOC\_124 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.329 CPSW\_NC\_EST\_FETCH\_LOC\_125 Register (Offset = 000321F4h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_125 is shown in [Table 9-361](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-361. CPSW\_NC\_EST\_FETCH\_LOC\_125 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.330 CPSW\_NC\_EST\_FETCH\_LOC\_126 Register (Offset = 000321F8h) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_126 is shown in [Table 9-362](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-362. CPSW\_NC\_EST\_FETCH\_LOC\_126 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location



### 9.6.331 CPSW\_NC\_EST\_FETCH\_LOC\_127 Register (Offset = 000321FCh) [Reset = 00000000h]

CPSW\_NC\_EST\_FETCH\_LOC\_127 is shown in [Table 9-363](#).

Return to the [Summary Table](#).

The Revision Register contains the ID and revision information.

**Table 9-363. CPSW\_NC\_EST\_FETCH\_LOC\_127 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-22	RESERVED	R	0h	
21-0	LOC	R/W	0h	RAM Location

### 9.6.332 CPSW\_CPDMA\_REGS\_CPDMA\_FH\_IDVER\_REG Register (Offset = 00034000h) [Reset = 0018010Ah]

CPSW\_CPDMA\_REGS\_CPDMA\_FH\_IDVER\_REG is shown in [Table 9-364](#).

Return to the [Summary Table](#).

CPDMA FHost IDVER

**Table 9-364. CPSW\_CPDMA\_REGS\_CPDMA\_FH\_IDVER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH_IDVER	R	0018010Ah	CPDMA FHost IDVER

### 9.6.333 CPSW\_CPDMA\_REGS\_CPDMA\_FH\_CONTROL\_REG Register (Offset = 00034004h) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_FH\_CONTROL\_REG is shown in [Table 9-365](#).

Return to the [Summary Table](#).

CPDMA FHost Control Register

**Table 9-365. CPSW\_CPDMA\_REGS\_CPDMA\_FH\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	FH_EN	R/W	0h	CPDMA FHost DMA Enable

### 9.6.334 CPSW\_CPDMA\_REGS\_CPDMA\_FH\_TEARDOWN\_REG Register (Offset = 00034008h) [Reset = XXXXXXXXh]

CPSW\_CPDMA\_REGS\_CPDMA\_FH\_TEARDOWN\_REG is shown in [Table 9-366](#).

Return to the [Summary Table](#).

CPDMA FHost Teardown Register

**Table 9-366. CPSW\_CPDMA\_REGS\_CPDMA\_FH\_TEARDOWN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	FH_TDN_RDY	R/W	0h	CPDMA FHost Teardown Ready
30-3	RESERVED	R	0h	
2-0	FH_TDN_CH	R/W	0h	CPDMA FHost Teardown Channel

### 9.6.335 CPSW\_CPDMA\_REGS\_CPDMA\_FH\_EOQ\_INT Register (Offset = 0003400Ch) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_FH\_EOQ\_INT is shown in [Table 9-367](#).

Return to the [Summary Table](#).

CPDMA FHost Interrupt on EOQ only Register

**Table 9-367. CPSW\_CPDMA\_REGS\_CPDMA\_FH\_EOQ\_INT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	FH_HW_TRIG_EN	R/W	0h	CPDMA FHost Hardware Trigger Control Enable
7-0	FH_EOQ_INT	R/W	0h	CPDMA FHost Interrupt on EOQ only

### 9.6.336 CPSW\_CPDMA\_REGS\_CPDMA\_TH\_IDVER\_REG Register (Offset = 00034010h) [Reset = 0018010Ah]

CPSW\_CPDMA\_REGS\_CPDMA\_TH\_IDVER\_REG is shown in [Table 9-368](#).

Return to the [Summary Table](#).

CPDMA THost IDVER

**Table 9-368. CPSW\_CPDMA\_REGS\_CPDMA\_TH\_IDVER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH_IDVER	R	0018010Ah	CPDMA THost IDVER

### 9.6.337 CPSW\_CPDMA\_REGS\_CPDMA\_TH\_CONTROL\_REG Register (Offset = 00034014h) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_TH\_CONTROL\_REG is shown in [Table 9-369](#).

Return to the [Summary Table](#).

CPDMA THost Control Register

**Table 9-369. CPSW\_CPDMA\_REGS\_CPDMA\_TH\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TH_EN	R/W	0h	CPDMA THost DMA Enable

### 9.6.338 CPSW\_CPDMA\_REGS\_CPDMA\_TH\_TEARDOWN\_REG Register (Offset = 00034018h) [Reset = XXXXXXXXh]

CPSW\_CPDMA\_REGS\_CPDMA\_TH\_TEARDOWN\_REG is shown in [Table 9-370](#).

Return to the [Summary Table](#).

CPDMA THost Teardown Register

**Table 9-370. CPSW\_CPDMA\_REGS\_CPDMA\_TH\_TEARDOWN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	TH_TDN_RDY	R/W	0h	CPDMA THost Teardown Ready
30-3	RESERVED	R	0h	
2-0	TH_TDN_CH	R/W	0h	CPDMA THost Teardown Channel



### 9.6.339 CPSW\_CPDMA\_REGS\_CPDMA\_SOFT\_RESET\_REG Register (Offset = 0003401Ch) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_SOFT\_RESET\_REG is shown in [Table 9-371](#).

Return to the [Summary Table](#).

CPDMA Soft Reset Register

**Table 9-371. CPSW\_CPDMA\_REGS\_CPDMA\_SOFT\_RESET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	SOFT_RESET	R/W	0h	CPDMA and CPSW Soft Reset Enable

### 9.6.340 CPSW\_CPDMA\_REGS\_CPDMA\_CONTROL\_REG Register (Offset = 00034020h) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_CONTROL\_REG is shown in [Table 9-372](#).

Return to the [Summary Table](#).

CPDMA Control Register

**Table 9-372. CPSW\_CPDMA\_REGS\_CPDMA\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R	0h	
8	FH_OWNERSHIP	R/W	0h	CPDMA FHost Ownership Write Bit Value
7	TH_CH_OVERRIDE	R/W	0h	CPDMA Channel Thread Override Enable
6	TH_TS_ENCAP	R/W	0h	CPDMA THost TimeStamp Encapsulated
5	TH_VLAN_ENCAP	R/W	0h	CPDMA THost VLAN Encapsulated
4	TH_CEF	R/W	0h	CPDMA THost Copy Error Frames
3	CMD_IDLE	R/W	0h	CPDMA Command Idle
2	TH_OFFLEN_BLOCK	R/W	0h	CPDMA THost Offset/Length Word Write Block
1	TH_OWNERSHIP	R/W	0h	CPDMA THost Ownership Write Bit Value
0	FH_PTYPE	R/W	0h	CPDMA FHost Queue Priority Type

### 9.6.341 CPSW\_CPDMA\_REGS\_CPDMA\_STATUS\_REG Register (Offset = 00034024h) [Reset = XX0X0XXXh]

CPSW\_CPDMA\_REGS\_CPDMA\_STATUS\_REG is shown in [Table 9-373](#).

Return to the [Summary Table](#).

CPDMA Status Register

**Table 9-373. CPSW\_CPDMA\_REGS\_CPDMA\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	IDLE	R	1h	CPDMA FHost Host Error Code
30-24	RESERVED	R	0h	
23-20	FH_HOST_ERROR_CODE	R	0h	CPDMA FHost Host Error Code
19	RESERVED	R	0h	
18-16	FH_ERR_CH	R	0h	CPDMA FHost Error Channel Number
15-12	TH_HOST_ERROR_CODE	R	0h	CPDMA THost Host Error Code
11	RESERVED	R	0h	
10-8	TH_ERR_CH	R	0h	CPDMA THost Error Channel Number
7-0	RESERVED	R	0h	

### 9.6.342 CPSW\_CPDMA\_REGS\_CPDMA\_TH\_BUFFER\_OFFSET\_REG Register (Offset = 00034028h) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_TH\_BUFFER\_OFFSET\_REG is shown in [Table 9-374](#).

Return to the [Summary Table](#).

CPDMA THost Buffer Offset Register

**Table 9-374. CPSW\_CPDMA\_REGS\_CPDMA\_TH\_BUFFER\_OFFSET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R	0h	
11-0	TH_BUFFER_OFFSET	R/W	0h	CPDMA THost Buffer Offset Register

### 9.6.343 CPSW\_CPDMA\_REGS\_CPDMA\_EMULATION\_CONTROL\_REG Register (Offset = 0003402Ch) [Reset = 00000000h]

CPSW\_CPDMA\_REGS\_CPDMA\_EMULATION\_CONTROL\_REG is shown in [Table 9-375](#).

Return to the [Summary Table](#).

CPDMA THost Buffer Offset Register

**Table 9-375. CPSW\_CPDMA\_REGS\_CPDMA\_EMULATION\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	FREE	R/W	0h	CPDMA THost Buffer Offset Register
0	SOFT	R/W	0h	CPDMA THost Buffer Offset Register

### 9.6.344 CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTSTAT\_RAW\_REG Register (Offset = 00034080h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTSTAT\_RAW\_REG is shown in [Table 9-376](#).

Return to the [Summary Table](#).

CPDMA FHost Interrupt Status RAW

**Table 9-376. CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTSTAT\_RAW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	FH7_PEND_RAW	R	0h	CPDMA FHost Channel 7 Interrupt Pending RAW
6	FH6_PEND_RAW	R	0h	CPDMA FHost Channel 6 Interrupt Pending RAW
5	FH5_PEND_RAW	R	0h	CPDMA FHost Channel 5 Interrupt Pending RAW
4	FH4_PEND_RAW	R	0h	CPDMA FHost Channel 4 Interrupt Pending RAW
3	FH3_PEND_RAW	R	0h	CPDMA FHost Channel 3 Interrupt Pending RAW
2	FH2_PEND_RAW	R	0h	CPDMA FHost Channel 2 Interrupt Pending RAW
1	FH1_PEND_RAW	R	0h	CPDMA FHost Channel 1 Interrupt Pending RAW
0	FH0_PEND_RAW	R	0h	CPDMA FHost Channel 0 Interrupt Pending RAW

### 9.6.345 CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTSTAT\_MASKED\_REG Register (Offset = 00034084h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTSTAT\_MASKED\_REG is shown in [Table 9-377](#).

Return to the [Summary Table](#).

CPDMA FHost Interrupt Status MASKED

**Table 9-377. CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTSTAT\_MASKED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	FH7_PEND_MASKED	R	0h	CPDMA FHost Channel 7 Interrupt Pending MASKED
6	FH6_PEND_MASKED	R	0h	CPDMA FHost Channel 6 Interrupt Pending MASKED
5	FH5_PEND_MASKED	R	0h	CPDMA FHost Channel 5 Interrupt Pending MASKED
4	FH4_PEND_MASKED	R	0h	CPDMA FHost Channel 4 Interrupt Pending MASKED
3	FH3_PEND_MASKED	R	0h	CPDMA FHost Channel 3 Interrupt Pending MASKED
2	FH2_PEND_MASKED	R	0h	CPDMA FHost Channel 2 Interrupt Pending MASKED
1	FH1_PEND_MASKED	R	0h	CPDMA FHost Channel 1 Interrupt Pending MASKED
0	FH0_PEND_MASKED	R	0h	CPDMA FHost Channel 0 Interrupt Pending MASKED

### 9.6.346 CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTMASK\_SET\_REG Register (Offset = 00034088h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTMASK\_SET\_REG is shown in [Table 9-378](#).

Return to the [Summary Table](#).

CPDMA FHost Interrupt Masked SET

**Table 9-378. CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTMASK\_SET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	FH7_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 7 Interrupt Pending MASKED Set
6	FH6_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 6 Interrupt Pending MASKED Set
5	FH5_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 5 Interrupt Pending MASKED Set
4	FH4_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 4 Interrupt Pending MASKED Set
3	FH3_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 3 Interrupt Pending MASKED Set
2	FH2_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 2 Interrupt Pending MASKED Set
1	FH1_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 1 Interrupt Pending MASKED Set
0	FH0_PEND_MASKED_SET	R/W1S	0h	CPDMA FHost Channel 0 Interrupt Pending MASKED Set



### 9.6.347 CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTMASK\_CLEAR\_REG Register (Offset = 0003408Ch) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTMASK\_CLEAR\_REG is shown in [Table 9-379](#).

Return to the [Summary Table](#).

CPDMA FHost Interrupt Masked CLR

**Table 9-379. CPSW\_CPDMA\_INT\_CPDMA\_FH\_INTMASK\_CLEAR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7	FH7_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 7 Interrupt Pending MASKED Clr
6	FH6_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 6 Interrupt Pending MASKED Clr
5	FH5_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 5 Interrupt Pending MASKED Clr
4	FH4_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 4 Interrupt Pending MASKED Clr
3	FH3_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 3 Interrupt Pending MASKED Clr
2	FH2_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 2 Interrupt Pending MASKED Clr
1	FH1_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 1 Interrupt Pending MASKED Clr
0	FH0_PEND_MASKED_CLR	R/W1C	0h	CPDMA FHost Channel 0 Interrupt Pending MASKED Clr

### 9.6.348 CPSW\_CPDMA\_INT\_CPDMA\_IN\_VECTOR\_REG Register (Offset = 00034090h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_IN\_VECTOR\_REG is shown in [Table 9-380](#).

Return to the [Summary Table](#).

CPDMA DMA IN Vector

**Table 9-380. CPSW\_CPDMA\_INT\_CPDMA\_IN\_VECTOR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DMA_IN_VECTOR	R	0h	CPDMA DMA IN Vector

### 9.6.349 CPSW\_CPDMA\_INT\_CPDMA\_EOI\_VECTOR\_REG Register (Offset = 00034094h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_EOI\_VECTOR\_REG is shown in [Table 9-381](#).

Return to the [Summary Table](#).

CPDMA DMA EOI Vector

**Table 9-381. CPSW\_CPDMA\_INT\_CPDMA\_EOI\_VECTOR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	DMA_EOI_VECTOR	R/W	0h	CPDMA DMA EOI Vector

### 9.6.350 CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTSTAT\_RAW\_REG Register (Offset = 000340A0h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTSTAT\_RAW\_REG is shown in [Table 9-382](#).

Return to the [Summary Table](#).

CPDMA Receive Interrupt Status RAW

**Table 9-382. CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTSTAT\_RAW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	TH7_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 7 Threshold Interrupt Pending RAW
14	TH6_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 6 Threshold Interrupt Pending RAW
13	TH5_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 5 Threshold Interrupt Pending RAW
12	TH4_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 4 Threshold Interrupt Pending RAW
11	TH3_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 3 Threshold Interrupt Pending RAW
10	TH2_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 2 Threshold Interrupt Pending RAW
9	TH1_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 1 Threshold Interrupt Pending RAW
8	TH0_THRESH_PEND_RAW	R	0h	CPDMA Receive Channel 0 Threshold Interrupt Pending RAW
7	TH7_PEND_RAW	R	0h	CPDMA Receive Channel 7 Interrupt Pending RAW
6	TH6_PEND_RAW	R	0h	CPDMA Receive Channel 6 Interrupt Pending RAW
5	TH5_PEND_RAW	R	0h	CPDMA Receive Channel 5 Interrupt Pending RAW
4	TH4_PEND_RAW	R	0h	CPDMA Receive Channel 4 Interrupt Pending RAW
3	TH3_PEND_RAW	R	0h	CPDMA Receive Channel 3 Interrupt Pending RAW
2	TH2_PEND_RAW	R	0h	CPDMA Receive Channel 2 Interrupt Pending RAW
1	TH1_PEND_RAW	R	0h	CPDMA Receive Channel 1 Interrupt Pending RAW
0	TH0_PEND_RAW	R	0h	CPDMA Receive Channel 0 Interrupt Pending RAW

### 9.6.351 CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTSTAT\_MASKED\_REG Register (Offset = 000340A4h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTSTAT\_MASKED\_REG is shown in [Table 9-383](#).

Return to the [Summary Table](#).

CPDMA Receive Interrupt Status MASKED

**Table 9-383. CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTSTAT\_MASKED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	TH7_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 7 Threshold Interrupt Pending MASKED
14	TH6_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 6 Threshold Interrupt Pending MASKED
13	TH5_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 5 Threshold Interrupt Pending MASKED
12	TH4_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 4 Threshold Interrupt Pending MASKED
11	TH3_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 3 Threshold Interrupt Pending MASKED
10	TH2_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 2 Threshold Interrupt Pending MASKED
9	TH1_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 1 Threshold Interrupt Pending MASKED
8	TH0_THRESH_PEND_MASKED	R	0h	CPDMA Receive Channel 0 Threshold Interrupt Pending MASKED
7	TH7_PEND_MASKED	R	0h	CPDMA Receive Channel 7 Interrupt Pending MASKED
6	TH6_PEND_MASKED	R	0h	CPDMA Receive Channel 6 Interrupt Pending MASKED
5	TH5_PEND_MASKED	R	0h	CPDMA Receive Channel 5 Interrupt Pending MASKED
4	TH4_PEND_MASKED	R	0h	CPDMA Receive Channel 4 Interrupt Pending MASKED
3	TH3_PEND_MASKED	R	0h	CPDMA Receive Channel 3 Interrupt Pending MASKED
2	TH2_PEND_MASKED	R	0h	CPDMA Receive Channel 2 Interrupt Pending MASKED
1	TH1_PEND_MASKED	R	0h	CPDMA Receive Channel 1 Interrupt Pending MASKED
0	TH0_PEND_MASKED	R	0h	CPDMA Receive Channel 0 Interrupt Pending MASKED

### 9.6.352 CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTMASK\_SET\_REG Register (Offset = 000340A8h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTMASK\_SET\_REG is shown in [Table 9-384](#).

Return to the [Summary Table](#).

CPDMA THost Interrupt Masked SET

**Table 9-384. CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTMASK\_SET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	TH7_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 7 Threshold Interrupt Pending SET
14	TH6_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 6 Threshold Interrupt Pending SET
13	TH5_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 5 Threshold Interrupt Pending SET
12	TH4_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 4 Threshold Interrupt Pending SET
11	TH3_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 3 Threshold Interrupt Pending SET
10	TH2_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 2 Threshold Interrupt Pending SET
9	TH1_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 1 Threshold Interrupt Pending SET
8	TH0_THRESH_PEND_M ASKED_SET	R/W1S	0h	CPDMA THost Channel 0 Threshold Interrupt Pending SET
7	TH7_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 7 Interrupt Pending SET
6	TH6_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 6 Interrupt Pending SET
5	TH5_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 5 Interrupt Pending SET
4	TH4_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 4 Interrupt Pending SET
3	TH3_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 3 Interrupt Pending SET
2	TH2_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 2 Interrupt Pending SET
1	TH1_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 1 Interrupt Pending SET
0	TH0_PEND_MASKED_SE T	R/W1S	0h	CPDMA THost Channel 0 Interrupt Pending SET

### 9.6.353 CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTMASK\_CLEAR\_REG Register (Offset = 000340ACh) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTMASK\_CLEAR\_REG is shown in [Table 9-385](#).

Return to the [Summary Table](#).

CPDMA THost Interrupt Masked CLR

**Table 9-385. CPSW\_CPDMA\_INT\_CPDMA\_TH\_INTMASK\_CLEAR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	TH7_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 7 Threshold Interrupt Pending CLR
14	TH6_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 6 Threshold Interrupt Pending CLR
13	TH5_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 5 Threshold Interrupt Pending CLR
12	TH4_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 4 Threshold Interrupt Pending CLR
11	TH3_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 3 Threshold Interrupt Pending CLR
10	TH2_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 2 Threshold Interrupt Pending CLR
9	TH1_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 1 Threshold Interrupt Pending CLR
8	TH0_THRESH_PEND_M ASKED_CLR	R/W1C	0h	CPDMA THost Channel 0 Threshold Interrupt Pending CLR
7	TH7_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 7 Interrupt Pending CLR
6	TH6_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 6 Interrupt Pending CLR
5	TH5_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 5 Interrupt Pending CLR
4	TH4_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 4 Interrupt Pending CLR
3	TH3_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 3 Interrupt Pending CLR
2	TH2_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 2 Interrupt Pending CLR
1	TH1_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 1 Interrupt Pending CLR
0	TH0_PEND_MASKED_CL R	R/W1C	0h	CPDMA THost Channel 0 Interrupt Pending CLR

### 9.6.354 CPSW\_CPDMA\_INT\_CPDMA\_INTSTAT\_RAW\_REG Register (Offset = 000340B0h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_INTSTAT\_RAW\_REG is shown in [Table 9-386](#).

Return to the [Summary Table](#).

CPDMA DMA Interrupt Status RAW

**Table 9-386. CPSW\_CPDMA\_INT\_CPDMA\_INTSTAT\_RAW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_PEND_RAW	R	0h	CPDMA HOST Interrupt Pending RAW
0	STAT_PEND_RAW	R	0h	CPDMA Statistics Interrupt Pending RAW



### 9.6.355 CPSW\_CPDMA\_INT\_CPDMA\_INTSTAT\_MASKED\_REG Register (Offset = 000340B4h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_INTSTAT\_MASKED\_REG is shown in [Table 9-387](#).

Return to the [Summary Table](#).

CPDMA DMA Interrupt Status MASKED

**Table 9-387. CPSW\_CPDMA\_INT\_CPDMA\_INTSTAT\_MASKED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_PEND	R	0h	CPDMA HOST Interrupt Pending MASKED
0	STAT_PEND	R	0h	CPDMA Statistics Interrupt Pending MASKED

### 9.6.356 CPSW\_CPDMA\_INT\_CPDMA\_INTMASK\_SET\_REG Register (Offset = 000340B8h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_INTMASK\_SET\_REG is shown in [Table 9-388](#).

Return to the [Summary Table](#).

CPDMA DMA Interrupt Status SET

**Table 9-388. CPSW\_CPDMA\_INT\_CPDMA\_INTMASK\_SET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_PEND_MASKED_SET	R/W1S	0h	CPDMA HOST Interrupt Masked SET
0	STAT_PEND_MASKED_SET	R/W1S	0h	CPDMA Statistics Interrupt Masked SET

### 9.6.357 CPSW\_CPDMA\_INT\_CPDMA\_INTMASK\_CLEAR\_REG Register (Offset = 000340BCh) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_INTMASK\_CLEAR\_REG is shown in [Table 9-389](#).

Return to the [Summary Table](#).

CPDMA DMA Interrupt Status CLR

**Table 9-389. CPSW\_CPDMA\_INT\_CPDMA\_INTMASK\_CLEAR\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	HOST_PEND_MASKED_CLR	R/W1S	0h	CPDMA HOST Interrupt Masked CLR
0	STAT_PEND_MASKED_CLR	R/W1S	0h	CPDMA Statistics Interrupt Masked CLR

### 9.6.358 CPSW\_CPDMA\_INT\_CPDMA\_TH0\_PENDTHRESH\_REG Register (Offset = 000340C0h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH0\_PENDTHRESH\_REG is shown in [Table 9-390](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-390. CPSW\_CPDMA\_INT\_CPDMA\_TH0\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH0_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.359 CPSW\_CPDMA\_INT\_CPDMA\_TH1\_PENDTHRESH\_REG Register (Offset = 000340C4h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH1\_PENDTHRESH\_REG is shown in [Table 9-391](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-391. CPSW\_CPDMA\_INT\_CPDMA\_TH1\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH1_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.360 CPSW\_CPDMA\_INT\_CPDMA\_TH2\_PENDTHRESH\_REG Register (Offset = 000340C8h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH2\_PENDTHRESH\_REG is shown in [Table 9-392](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-392. CPSW\_CPDMA\_INT\_CPDMA\_TH2\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH2_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.361 CPSW\_CPDMA\_INT\_CPDMA\_TH3\_PENDTHRESH\_REG Register (Offset = 000340CCh) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH3\_PENDTHRESH\_REG is shown in [Table 9-393](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-393. CPSW\_CPDMA\_INT\_CPDMA\_TH3\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH3_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.362 CPSW\_CPDMA\_INT\_CPDMA\_TH4\_PENDTHRESH\_REG Register (Offset = 000340D0h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH4\_PENDTHRESH\_REG is shown in [Table 9-394](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-394. CPSW\_CPDMA\_INT\_CPDMA\_TH4\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH4_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register



### 9.6.363 CPSW\_CPDMA\_INT\_CPDMA\_TH5\_PENDTHRESH\_REG Register (Offset = 000340D4h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH5\_PENDTHRESH\_REG is shown in [Table 9-395](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-395. CPSW\_CPDMA\_INT\_CPDMA\_TH5\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH5_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.364 CPSW\_CPDMA\_INT\_CPDMA\_TH6\_PENDTHRESH\_REG Register (Offset = 000340D8h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH6\_PENDTHRESH\_REG is shown in [Table 9-396](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-396. CPSW\_CPDMA\_INT\_CPDMA\_TH6\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH6_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.365 CPSW\_CPDMA\_INT\_CPDMA\_TH7\_PENDTHRESH\_REG Register (Offset = 000340DCh) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH7\_PENDTHRESH\_REG is shown in [Table 9-397](#).

Return to the [Summary Table](#).

CPDMA THost Threshold Pending Register

**Table 9-397. CPSW\_CPDMA\_INT\_CPDMA\_TH7\_PENDTHRESH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TH7_PENDTHRESH	R/W	0h	CPDMA THost Threshold Pending Register

### 9.6.366 CPSW\_CPDMA\_INT\_CPDMA\_TH0\_FREEBUFFER\_REG Register (Offset = 000340E0h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH0\_FREEBUFFER\_REG is shown in [Table 9-398](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-398. CPSW\_CPDMA\_INT\_CPDMA\_TH0\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH0_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.367 CPSW\_CPDMA\_INT\_CPDMA\_TH1\_FREEBUFFER\_REG Register (Offset = 000340E4h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH1\_FREEBUFFER\_REG is shown in [Table 9-399](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-399. CPSW\_CPDMA\_INT\_CPDMA\_TH1\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH1_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.368 CPSW\_CPDMA\_INT\_CPDMA\_TH2\_FREEBUFFER\_REG Register (Offset = 000340E8h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH2\_FREEBUFFER\_REG is shown in [Table 9-400](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-400. CPSW\_CPDMA\_INT\_CPDMA\_TH2\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH2_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.369 CPSW\_CPDMA\_INT\_CPDMA\_TH3\_FREEBUFFER\_REG Register (Offset = 000340ECh) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH3\_FREEBUFFER\_REG is shown in [Table 9-401](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-401. CPSW\_CPDMA\_INT\_CPDMA\_TH3\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH3_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.370 CPSW\_CPDMA\_INT\_CPDMA\_TH4\_FREEBUFFER\_REG Register (Offset = 000340F0h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH4\_FREEBUFFER\_REG is shown in [Table 9-402](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-402. CPSW\_CPDMA\_INT\_CPDMA\_TH4\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH4_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register



### 9.6.371 CPSW\_CPDMA\_INT\_CPDMA\_TH5\_FREEBUFFER\_REG Register (Offset = 000340F4h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH5\_FREEBUFFER\_REG is shown in [Table 9-403](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-403. CPSW\_CPDMA\_INT\_CPDMA\_TH5\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH5_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.372 CPSW\_CPDMA\_INT\_CPDMA\_TH6\_FREEBUFFER\_REG Register (Offset = 000340F8h) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH6\_FREEBUFFER\_REG is shown in [Table 9-404](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-404. CPSW\_CPDMA\_INT\_CPDMA\_TH6\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH6_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.373 CPSW\_CPDMA\_INT\_CPDMA\_TH7\_FREEBUFFER\_REG Register (Offset = 000340FCh) [Reset = 00000000h]

CPSW\_CPDMA\_INT\_CPDMA\_TH7\_FREEBUFFER\_REG is shown in [Table 9-405](#).

Return to the [Summary Table](#).

CPDMA THost Free Buffer Count Register

**Table 9-405. CPSW\_CPDMA\_INT\_CPDMA\_TH7\_FREEBUFFER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	RESERVED	R	0h	
14-0	TH7_FREEBUFFER	R/W	0h	CPDMA THost Free Buffer Count Register

### 9.6.374 CPSW\_CPDMA\_SRAM\_CPDMA\_FH0\_HDP\_REG Register (Offset = 00034200h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH0\_HDP\_REG is shown in [Table 9-406](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 0 Head Descriptor Pointer

**Table 9-406. CPSW\_CPDMA\_SRAM\_CPDMA\_FH0\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH0_HDP	R/W	0h	CPDMA FHost Channel 0 Head Descriptor Pointer

### 9.6.375 CPSW\_CPDMA\_SRAM\_CPDMA\_FH1\_HDP\_REG Register (Offset = 00034204h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH1\_HDP\_REG is shown in [Table 9-407](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 1 Head Descriptor Pointer

**Table 9-407. CPSW\_CPDMA\_SRAM\_CPDMA\_FH1\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH1_HDP	R/W	0h	CPDMA FHost Channel 1 Head Descriptor Pointer

### 9.6.376 CPSW\_CPDMA\_SRAM\_CPDMA\_FH2\_HDP\_REG Register (Offset = 00034208h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH2\_HDP\_REG is shown in [Table 9-408](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 2 Head Descriptor Pointer

**Table 9-408. CPSW\_CPDMA\_SRAM\_CPDMA\_FH2\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH2_HDP	R/W	0h	CPDMA FHost Channel 2 Head Descriptor Pointer

### 9.6.377 CPSW\_CPDMA\_SRAM\_CPDMA\_FH3\_HDP\_REG Register (Offset = 0003420Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH3\_HDP\_REG is shown in [Table 9-409](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 3 Head Descriptor Pointer

**Table 9-409. CPSW\_CPDMA\_SRAM\_CPDMA\_FH3\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH3_HDP	R/W	0h	CPDMA FHost Channel 3 Head Descriptor Pointer

### 9.6.378 CPSW\_CPDMA\_SRAM\_CPDMA\_FH4\_HDP\_REG Register (Offset = 00034210h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH4\_HDP\_REG is shown in [Table 9-410](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 4 Head Descriptor Pointer

**Table 9-410. CPSW\_CPDMA\_SRAM\_CPDMA\_FH4\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH4_HDP	R/W	0h	CPDMA FHost Channel 4 Head Descriptor Pointer



### 9.6.379 CPSW\_CPDMA\_SRAM\_CPDMA\_FH5\_HDP\_REG Register (Offset = 00034214h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH5\_HDP\_REG is shown in [Table 9-411](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 5 Head Descriptor Pointer

**Table 9-411. CPSW\_CPDMA\_SRAM\_CPDMA\_FH5\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH5_HDP	R/W	0h	CPDMA FHost Channel 5 Head Descriptor Pointer

### 9.6.380 CPSW\_CPDMA\_SRAM\_CPDMA\_FH6\_HDP\_REG Register (Offset = 00034218h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH6\_HDP\_REG is shown in [Table 9-412](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 6 Head Descriptor Pointer

**Table 9-412. CPSW\_CPDMA\_SRAM\_CPDMA\_FH6\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH6_HDP	R/W	0h	CPDMA FHost Channel 6 Head Descriptor Pointer

### 9.6.381 CPSW\_CPDMA\_SRAM\_CPDMA\_FH7\_HDP\_REG Register (Offset = 0003421Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH7\_HDP\_REG is shown in [Table 9-413](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 7 Head Descriptor Pointer

**Table 9-413. CPSW\_CPDMA\_SRAM\_CPDMA\_FH7\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH7_HDP	R/W	0h	CPDMA FHost Channel 7 Head Descriptor Pointer

### 9.6.382 CPSW\_CPDMA\_SRAM\_CPDMA\_TH0\_HDP\_REG Register (Offset = 00034220h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH0\_HDP\_REG is shown in [Table 9-414](#).

Return to the [Summary Table](#).

CPDMA THost Channel 0 Head Descriptor Pointer

**Table 9-414. CPSW\_CPDMA\_SRAM\_CPDMA\_TH0\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH0_HDP	R/W	0h	CPDMA THost Channel 0 Head Descriptor Pointer

### 9.6.383 CPSW\_CPDMA\_SRAM\_CPDMA\_TH1\_HDP\_REG Register (Offset = 00034224h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH1\_HDP\_REG is shown in [Table 9-415](#).

Return to the [Summary Table](#).

CPDMA THost Channel 1 Head Descriptor Pointer

**Table 9-415. CPSW\_CPDMA\_SRAM\_CPDMA\_TH1\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH1_HDP	R/W	0h	CPDMA THost Channel 1 Head Descriptor Pointer

### 9.6.384 CPSW\_CPDMA\_SRAM\_CPDMA\_TH2\_HDP\_REG Register (Offset = 00034228h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH2\_HDP\_REG is shown in [Table 9-416](#).

Return to the [Summary Table](#).

CPDMA THost Channel 2 Head Descriptor Pointer

**Table 9-416. CPSW\_CPDMA\_SRAM\_CPDMA\_TH2\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH2_HDP	R/W	0h	CPDMA THost Channel 2 Head Descriptor Pointer

### 9.6.385 CPSW\_CPDMA\_SRAM\_CPDMA\_TH3\_HDP\_REG Register (Offset = 0003422Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH3\_HDP\_REG is shown in [Table 9-417](#).

Return to the [Summary Table](#).

CPDMA THost Channel 3 Head Descriptor Pointer

**Table 9-417. CPSW\_CPDMA\_SRAM\_CPDMA\_TH3\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH3_HDP	R/W	0h	CPDMA THost Channel 3 Head Descriptor Pointer

### 9.6.386 CPSW\_CPDMA\_SRAM\_CPDMA\_TH4\_HDP\_REG Register (Offset = 00034230h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH4\_HDP\_REG is shown in [Table 9-418](#).

Return to the [Summary Table](#).

CPDMA THost Channel 4 Head Descriptor Pointer

**Table 9-418. CPSW\_CPDMA\_SRAM\_CPDMA\_TH4\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH4_HDP	R/W	0h	CPDMA THost Channel 4 Head Descriptor Pointer



### 9.6.387 CPSW\_CPDMA\_SRAM\_CPDMA\_TH5\_HDP\_REG Register (Offset = 00034234h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH5\_HDP\_REG is shown in [Table 9-419](#).

Return to the [Summary Table](#).

CPDMA THost Channel 5 Head Descriptor Pointer

**Table 9-419. CPSW\_CPDMA\_SRAM\_CPDMA\_TH5\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH5_HDP	R/W	0h	CPDMA THost Channel 5 Head Descriptor Pointer

### 9.6.388 CPSW\_CPDMA\_SRAM\_CPDMA\_TH6\_HDP\_REG Register (Offset = 00034238h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH6\_HDP\_REG is shown in [Table 9-420](#).

Return to the [Summary Table](#).

CPDMA THost Channel 6 Head Descriptor Pointer

**Table 9-420. CPSW\_CPDMA\_SRAM\_CPDMA\_TH6\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH6_HDP	R/W	0h	CPDMA THost Channel 6 Head Descriptor Pointer

### 9.6.389 CPSW\_CPDMA\_SRAM\_CPDMA\_TH7\_HDP\_REG Register (Offset = 0003423Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH7\_HDP\_REG is shown in [Table 9-421](#).

Return to the [Summary Table](#).

CPDMA THost Channel 7 Head Descriptor Pointer

**Table 9-421. CPSW\_CPDMA\_SRAM\_CPDMA\_TH7\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH7_HDP	R/W	0h	CPDMA THost Channel 7 Head Descriptor Pointer

### 9.6.390 CPSW\_CPDMA\_SRAM\_CPDMA\_FH0\_CP\_REG Register (Offset = 00034240h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH0\_CP\_REG is shown in [Table 9-422](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 0 Completion Pointer

**Table 9-422. CPSW\_CPDMA\_SRAM\_CPDMA\_FH0\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH0_CP	R/W	0h	CPDMA FHost Channel 0 Completion Pointer

### 9.6.391 CPSW\_CPDMA\_SRAM\_CPDMA\_FH1\_CP\_REG Register (Offset = 00034244h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH1\_CP\_REG is shown in [Table 9-423](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 1 Completion Pointer

**Table 9-423. CPSW\_CPDMA\_SRAM\_CPDMA\_FH1\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH1_CP	R/W	0h	CPDMA FHost Channel 1 Completion Pointer

### 9.6.392 CPSW\_CPDMA\_SRAM\_CPDMA\_FH2\_CP\_REG Register (Offset = 00034248h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH2\_CP\_REG is shown in [Table 9-424](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 2 Completion Pointer

**Table 9-424. CPSW\_CPDMA\_SRAM\_CPDMA\_FH2\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH2_CP	R/W	0h	CPDMA FHost Channel 2 Completion Pointer

### 9.6.393 CPSW\_CPDMA\_SRAM\_CPDMA\_FH3\_CP\_REG Register (Offset = 0003424Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH3\_CP\_REG is shown in [Table 9-425](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 3 Completion Pointer

**Table 9-425. CPSW\_CPDMA\_SRAM\_CPDMA\_FH3\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH3_CP	R/W	0h	CPDMA FHost Channel 3 Completion Pointer

### 9.6.394 CPSW\_CPDMA\_SRAM\_CPDMA\_FH4\_CP\_REG Register (Offset = 00034250h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH4\_CP\_REG is shown in [Table 9-426](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 4 Completion Pointer

**Table 9-426. CPSW\_CPDMA\_SRAM\_CPDMA\_FH4\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH4_CP	R/W	0h	CPDMA FHost Channel 4 Completion Pointer



### 9.6.395 CPSW\_CPDMA\_SRAM\_CPDMA\_FH5\_CP\_REG Register (Offset = 00034254h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH5\_CP\_REG is shown in [Table 9-427](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 5 Completion Pointer

**Table 9-427. CPSW\_CPDMA\_SRAM\_CPDMA\_FH5\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH5_CP	R/W	0h	CPDMA FHost Channel 5 Completion Pointer

### 9.6.396 CPSW\_CPDMA\_SRAM\_CPDMA\_FH6\_CP\_REG Register (Offset = 00034258h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH6\_CP\_REG is shown in [Table 9-428](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 6 Completion Pointer

**Table 9-428. CPSW\_CPDMA\_SRAM\_CPDMA\_FH6\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH6_CP	R/W	0h	CPDMA FHost Channel 6 Completion Pointer

### 9.6.397 CPSW\_CPDMA\_SRAM\_CPDMA\_FH7\_CP\_REG Register (Offset = 0003425Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_FH7\_CP\_REG is shown in [Table 9-429](#).

Return to the [Summary Table](#).

CPDMA FHost Channel 7 Completion Pointer

**Table 9-429. CPSW\_CPDMA\_SRAM\_CPDMA\_FH7\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FH7_CP	R/W	0h	CPDMA FHost Channel 7 Completion Pointer

### 9.6.398 CPSW\_CPDMA\_SRAM\_CPDMA\_TH0\_CP\_REG Register (Offset = 00034260h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH0\_CP\_REG is shown in [Table 9-430](#).

Return to the [Summary Table](#).

CPDMA THost Channel 0 Completion Pointer

**Table 9-430. CPSW\_CPDMA\_SRAM\_CPDMA\_TH0\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH0_CP	R/W	0h	CPDMA THost Channel 0 Completion Pointer

### 9.6.399 CPSW\_CPDMA\_SRAM\_CPDMA\_TH1\_CP\_REG Register (Offset = 00034264h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH1\_CP\_REG is shown in [Table 9-431](#).

Return to the [Summary Table](#).

CPDMA THost Channel 1 Completion Pointer

**Table 9-431. CPSW\_CPDMA\_SRAM\_CPDMA\_TH1\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH1_CP	R/W	0h	CPDMA THost Channel 1 Completion Pointer

### 9.6.400 CPSW\_CPDMA\_SRAM\_CPDMA\_TH2\_CP\_REG Register (Offset = 00034268h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH2\_CP\_REG is shown in [Table 9-432](#).

Return to the [Summary Table](#).

CPDMA THost Channel 2 Completion Pointer

**Table 9-432. CPSW\_CPDMA\_SRAM\_CPDMA\_TH2\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH2_CP	R/W	0h	CPDMA THost Channel 2 Completion Pointer

### 9.6.401 CPSW\_CPDMA\_SRAM\_CPDMA\_TH3\_CP\_REG Register (Offset = 0003426Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH3\_CP\_REG is shown in [Table 9-433](#).

Return to the [Summary Table](#).

CPDMA THost Channel 3 Completion Pointer

**Table 9-433. CPSW\_CPDMA\_SRAM\_CPDMA\_TH3\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH3_CP	R/W	0h	CPDMA THost Channel 3 Completion Pointer

### 9.6.402 CPSW\_CPDMA\_SRAM\_CPDMA\_TH4\_CP\_REG Register (Offset = 00034270h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH4\_CP\_REG is shown in [Table 9-434](#).

Return to the [Summary Table](#).

CPDMA THost Channel 4 Completion Pointer

**Table 9-434. CPSW\_CPDMA\_SRAM\_CPDMA\_TH4\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH4_CP	R/W	0h	CPDMA THost Channel 4 Completion Pointer



### 9.6.403 CPSW\_CPDMA\_SRAM\_CPDMA\_TH5\_CP\_REG Register (Offset = 00034274h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH5\_CP\_REG is shown in [Table 9-435](#).

Return to the [Summary Table](#).

CPDMA THost Channel 5 Completion Pointer

**Table 9-435. CPSW\_CPDMA\_SRAM\_CPDMA\_TH5\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH5_CP	R/W	0h	CPDMA THost Channel 5 Completion Pointer

#### 9.6.404 CPSW\_CPDMA\_SRAM\_CPDMA\_TH6\_CP\_REG Register (Offset = 00034278h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH6\_CP\_REG is shown in [Table 9-436](#).

Return to the [Summary Table](#).

CPDMA THost Channel 6 Completion Pointer

**Table 9-436. CPSW\_CPDMA\_SRAM\_CPDMA\_TH6\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH6_CP	R/W	0h	CPDMA THost Channel 6 Completion Pointer

### 9.6.405 CPSW\_CPDMA\_SRAM\_CPDMA\_TH7\_CP\_REG Register (Offset = 0003427Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_CPDMA\_TH7\_CP\_REG is shown in [Table 9-437](#).

Return to the [Summary Table](#).

CPDMA THost Channel 7 Completion Pointer

**Table 9-437. CPSW\_CPDMA\_SRAM\_CPDMA\_TH7\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TH7_CP	R/W	0h	CPDMA THost Channel 7 Completion Pointer

### 9.6.406 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH0\_HDP\_REG Register (Offset = 00034300h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH0\_HDP\_REG is shown in [Table 9-438](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 0 Head Descriptor Pointer

**Table 9-438. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH0\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH0_HDP	R/W	0h	Test CPDMA FHost Channel 0 Head Descriptor Pointer

### 9.6.407 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH1\_HDP\_REG Register (Offset = 00034304h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH1\_HDP\_REG is shown in [Table 9-439](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 1 Head Descriptor Pointer

**Table 9-439. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH1\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH1_HDP	R/W	0h	Test CPDMA FHost Channel 1 Head Descriptor Pointer

### 9.6.408 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH2\_HDP\_REG Register (Offset = 00034308h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH2\_HDP\_REG is shown in [Table 9-440](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 2 Head Descriptor Pointer

**Table 9-440. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH2\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH2_HDP	R/W	0h	Test CPDMA FHost Channel 2 Head Descriptor Pointer

### 9.6.409 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH3\_HDP\_REG Register (Offset = 0003430Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH3\_HDP\_REG is shown in [Table 9-441](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 3 Head Descriptor Pointer

**Table 9-441. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH3\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH3_HDP	R/W	0h	Test CPDMA FHost Channel 3 Head Descriptor Pointer

### 9.6.410 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH4\_HDP\_REG Register (Offset = 00034310h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH4\_HDP\_REG is shown in [Table 9-442](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 4 Head Descriptor Pointer

**Table 9-442. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH4\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH4_HDP	R/W	0h	Test CPDMA FHost Channel 4 Head Descriptor Pointer



### 9.6.411 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH5\_HDP\_REG Register (Offset = 00034314h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH5\_HDP\_REG is shown in [Table 9-443](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 5 Head Descriptor Pointer

**Table 9-443. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH5\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH5_HDP	R/W	0h	Test CPDMA FHost Channel 5 Head Descriptor Pointer

### 9.6.412 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH6\_HDP\_REG Register (Offset = 00034318h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH6\_HDP\_REG is shown in [Table 9-444](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 6 Head Descriptor Pointer

**Table 9-444. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH6\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH6_HDP	R/W	0h	Test CPDMA FHost Channel 6 Head Descriptor Pointer

### 9.6.413 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH7\_HDP\_REG Register (Offset = 0003431Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH7\_HDP\_REG is shown in [Table 9-445](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 7 Head Descriptor Pointer

**Table 9-445. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH7\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH7_HDP	R/W	0h	Test CPDMA FHost Channel 7 Head Descriptor Pointer

### 9.6.414 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH0\_HDP\_REG Register (Offset = 00034320h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH0\_HDP\_REG is shown in [Table 9-446](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 0 Head Descriptor Pointer

**Table 9-446. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH0\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH0_HDP	R/W	0h	Test CPDMA THost Channel 0 Head Descriptor Pointer

### 9.6.415 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH1\_HDP\_REG Register (Offset = 00034324h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH1\_HDP\_REG is shown in [Table 9-447](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 1 Head Descriptor Pointer

**Table 9-447. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH1\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH1_HDP	R/W	0h	Test CPDMA THost Channel 1 Head Descriptor Pointer

### 9.6.416 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH2\_HDP\_REG Register (Offset = 00034328h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH2\_HDP\_REG is shown in [Table 9-448](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 2 Head Descriptor Pointer

**Table 9-448. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH2\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH2_HDP	R/W	0h	Test CPDMA THost Channel 2 Head Descriptor Pointer

### 9.6.417 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH3\_HDP\_REG Register (Offset = 0003432Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH3\_HDP\_REG is shown in [Table 9-449](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 3 Head Descriptor Pointer

**Table 9-449. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH3\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH3_HDP	R/W	0h	Test CPDMA THost Channel 3 Head Descriptor Pointer

### 9.6.418 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH4\_HDP\_REG Register (Offset = 00034330h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH4\_HDP\_REG is shown in [Table 9-450](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 4 Head Descriptor Pointer

**Table 9-450. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH4\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH4_HDP	R/W	0h	Test CPDMA THost Channel 4 Head Descriptor Pointer



### 9.6.419 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH5\_HDP\_REG Register (Offset = 00034334h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH5\_HDP\_REG is shown in [Table 9-451](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 5 Head Descriptor Pointer

**Table 9-451. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH5\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH5_HDP	R/W	0h	Test CPDMA THost Channel 5 Head Descriptor Pointer

### 9.6.420 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH6\_HDP\_REG Register (Offset = 00034338h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH6\_HDP\_REG is shown in [Table 9-452](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 6 Head Descriptor Pointer

**Table 9-452. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH6\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH6_HDP	R/W	0h	Test CPDMA THost Channel 6 Head Descriptor Pointer

### 9.6.421 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH7\_HDP\_REG Register (Offset = 0003433Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH7\_HDP\_REG is shown in [Table 9-453](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 7 Head Descriptor Pointer

**Table 9-453. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH7\_HDP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH7_HDP	R/W	0h	Test CPDMA THost Channel 7 Head Descriptor Pointer

### 9.6.422 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH0\_CP\_REG Register (Offset = 00034340h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH0\_CP\_REG is shown in [Table 9-454](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 0 Completion Pointer

**Table 9-454. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH0\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH0_CP	R/W	0h	Test CPDMA FHost Channel 0 Completion Pointer

### 9.6.423 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH1\_CP\_REG Register (Offset = 00034344h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH1\_CP\_REG is shown in [Table 9-455](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 1 Completion Pointer

**Table 9-455. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH1\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH1_CP	R/W	0h	Test CPDMA FHost Channel 1 Completion Pointer

### 9.6.424 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH2\_CP\_REG Register (Offset = 00034348h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH2\_CP\_REG is shown in [Table 9-456](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 2 Completion Pointer

**Table 9-456. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH2\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH2_CP	R/W	0h	Test CPDMA FHost Channel 2 Completion Pointer

### 9.6.425 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH3\_CP\_REG Register (Offset = 0003434Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH3\_CP\_REG is shown in [Table 9-457](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 3 Completion Pointer

**Table 9-457. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH3\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH3_CP	R/W	0h	Test CPDMA FHost Channel 3 Completion Pointer

### 9.6.426 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH4\_CP\_REG Register (Offset = 00034350h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH4\_CP\_REG is shown in [Table 9-458](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 4 Completion Pointer

**Table 9-458. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH4\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH4_CP	R/W	0h	Test CPDMA FHost Channel 4 Completion Pointer



### 9.6.427 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH5\_CP\_REG Register (Offset = 00034354h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH5\_CP\_REG is shown in [Table 9-459](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 5 Completion Pointer

**Table 9-459. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH5\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH5_CP	R/W	0h	Test CPDMA FHost Channel 5 Completion Pointer

### 9.6.428 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH6\_CP\_REG Register (Offset = 00034358h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH6\_CP\_REG is shown in [Table 9-460](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 6 Completion Pointer

**Table 9-460. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH6\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH6_CP	R/W	0h	Test CPDMA FHost Channel 6 Completion Pointer

### 9.6.429 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH7\_CP\_REG Register (Offset = 0003435Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH7\_CP\_REG is shown in [Table 9-461](#).

Return to the [Summary Table](#).

Test CPDMA FHost Channel 7 Completion Pointer

**Table 9-461. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_FH7\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_FH7_CP	R/W	0h	Test CPDMA FHost Channel 7 Completion Pointer

### 9.6.430 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH0\_CP\_REG Register (Offset = 00034360h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH0\_CP\_REG is shown in [Table 9-462](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 0 Completion Pointer

**Table 9-462. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH0\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH0_CP	R/W	0h	Test CPDMA THost Channel 0 Completion Pointer

### 9.6.431 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH1\_CP\_REG Register (Offset = 00034364h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH1\_CP\_REG is shown in [Table 9-463](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 1 Completion Pointer

**Table 9-463. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH1\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH1_CP	R/W	0h	Test CPDMA THost Channel 1 Completion Pointer

### 9.6.432 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH2\_CP\_REG Register (Offset = 00034368h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH2\_CP\_REG is shown in [Table 9-464](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 2 Completion Pointer

**Table 9-464. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH2\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH2_CP	R/W	0h	Test CPDMA THost Channel 2 Completion Pointer

### 9.6.433 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH3\_CP\_REG Register (Offset = 0003436Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH3\_CP\_REG is shown in [Table 9-465](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 3 Completion Pointer

**Table 9-465. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH3\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH3_CP	R/W	0h	Test CPDMA THost Channel 3 Completion Pointer

### 9.6.434 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH4\_CP\_REG Register (Offset = 00034370h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH4\_CP\_REG is shown in [Table 9-466](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 4 Completion Pointer

**Table 9-466. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH4\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH4_CP	R/W	0h	Test CPDMA THost Channel 4 Completion Pointer



### 9.6.435 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH5\_CP\_REG Register (Offset = 00034374h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH5\_CP\_REG is shown in [Table 9-467](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 5 Completion Pointer

**Table 9-467. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH5\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH5_CP	R/W	0h	Test CPDMA THost Channel 5 Completion Pointer

### 9.6.436 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH6\_CP\_REG Register (Offset = 00034378h) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH6\_CP\_REG is shown in [Table 9-468](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 6 Completion Pointer

**Table 9-468. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH6\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH6_CP	R/W	0h	Test CPDMA THost Channel 6 Completion Pointer

### 9.6.437 CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH7\_CP\_REG Register (Offset = 0003437Ch) [Reset = 00000000h]

CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH7\_CP\_REG is shown in [Table 9-469](#).

Return to the [Summary Table](#).

Test CPDMA THost Channel 7 Completion Pointer

**Table 9-469. CPSW\_CPDMA\_SRAM\_TEST\_CPDMA\_TH7\_CP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TEST_TH7_CP	R/W	0h	Test CPDMA THost Channel 7 Completion Pointer

### 9.6.438 CPSW\_NC\_STAT\_0\_RXGOODFRAMES Register (Offset = 0003A000h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXGOODFRAMES is shown in [Table 9-470](#).

Return to the [Summary Table](#).

Total number of good frames received

**Table 9-470. CPSW\_NC\_STAT\_0\_RXGOODFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good frames received

### 9.6.439 CPSW\_NC\_STAT\_0\_RXBROADCASTFRAMES Register (Offset = 0003A004h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXBROADCASTFRAMES is shown in [Table 9-471](#).

Return to the [Summary Table](#).

Total number of good broadcast frames received

**Table 9-471. CPSW\_NC\_STAT\_0\_RXBROADCASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good broadcast frames received

#### 9.6.440 CPSW\_NC\_STAT\_0\_RXMULTICASTFRAMES Register (Offset = 0003A008h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXMULTICASTFRAMES is shown in [Table 9-472](#).

Return to the [Summary Table](#).

Total number of good multicast frames received

**Table 9-472. CPSW\_NC\_STAT\_0\_RXMULTICASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good multicast frames received

#### 9.6.441 CPSW\_NC\_STAT\_0\_RXCRCERRORS Register (Offset = 0003A010h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXCRCERRORS is shown in [Table 9-473](#).

Return to the [Summary Table](#).

Total number of CRC errors frames received

**Table 9-473. CPSW\_NC\_STAT\_0\_RXCRCERRORS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of CRC errors frames received

### 9.6.442 CPSW\_NC\_STAT\_0\_RXOVERSIZEDFRAMES Register (Offset = 0003A018h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXOVERSIZEDFRAMES is shown in [Table 9-474](#).

Return to the [Summary Table](#).

Total number of oversized frames received

**Table 9-474. CPSW\_NC\_STAT\_0\_RXOVERSIZEDFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of oversized frames received



#### 9.6.443 CPSW\_NC\_STAT\_0\_RXUNDERSIZEDFRAMES Register (Offset = 0003A020h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXUNDERSIZEDFRAMES is shown in [Table 9-475](#).

Return to the [Summary Table](#).

Total number of undersized frames received

**Table 9-475. CPSW\_NC\_STAT\_0\_RXUNDERSIZEDFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of undersized frames received

#### 9.6.444 CPSW\_NC\_STAT\_0\_RXFRAGMENTS Register (Offset = 0003A024h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXFRAGMENTS is shown in [Table 9-476](#).

Return to the [Summary Table](#).

Total number of fragmented frames received

**Table 9-476. CPSW\_NC\_STAT\_0\_RXFRAGMENTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of fragmented frames received

### 9.6.445 CPSW\_NC\_STAT\_0\_ALE\_DROP Register (Offset = 0003A028h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_DROP is shown in [Table 9-477](#).

Return to the [Summary Table](#).

Total number of frames dropped by the ALE

**Table 9-477. CPSW\_NC\_STAT\_0\_ALE\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames dropped by the ALE

### 9.6.446 CPSW\_NC\_STAT\_0\_ALE\_OVERRUN\_DROP Register (Offset = 0003A02Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_OVERRUN\_DROP is shown in [Table 9-478](#).

Return to the [Summary Table](#).

Total number of overrun frames dropped by the ALE

**Table 9-478. CPSW\_NC\_STAT\_0\_ALE\_OVERRUN\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of overrun frames dropped by the ALE

### 9.6.447 CPSW\_NC\_STAT\_0\_RXOCTETS Register (Offset = 0003A030h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RXOCTETS is shown in [Table 9-479](#).

Return to the [Summary Table](#).

Total number of received bytes in good frames

**Table 9-479. CPSW\_NC\_STAT\_0\_RXOCTETS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of received bytes in good frames

### 9.6.448 CPSW\_NC\_STAT\_0\_TXGOODFRAMES Register (Offset = 0003A034h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TXGOODFRAMES is shown in [Table 9-480](#).

Return to the [Summary Table](#).

Total number of good frames transmitted

**Table 9-480. CPSW\_NC\_STAT\_0\_TXGOODFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good frames transmitted

#### 9.6.449 CPSW\_NC\_STAT\_0\_TXBROADCASTFRAMES Register (Offset = 0003A038h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TXBROADCASTFRAMES is shown in [Table 9-481](#).

Return to the [Summary Table](#).

Total number of good broadcast frames transmitted

**Table 9-481. CPSW\_NC\_STAT\_0\_TXBROADCASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good broadcast frames transmitted

### 9.6.450 CPSW\_NC\_STAT\_0\_TXMULTICASTFRAMES Register (Offset = 0003A03Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TXMULTICASTFRAMES is shown in [Table 9-482](#).

Return to the [Summary Table](#).

Total number of good multicast frames transmitted

**Table 9-482. CPSW\_NC\_STAT\_0\_TXMULTICASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good multicast frames transmitted



### 9.6.451 CPSW\_NC\_STAT\_0\_TXSINGLECOLLFRAMES Register (Offset = 0003A04Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TXSINGLECOLLFRAMES is shown in [Table 9-483](#).

Return to the [Summary Table](#).

Total number of transmitted frames experiencing a single collision

**Table 9-483. CPSW\_NC\_STAT\_0\_TXSINGLECOLLFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames experiencing a single collision

### 9.6.452 CPSW\_NC\_STAT\_0\_TXMULTCOLLFRAMES Register (Offset = 0003A050h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TXMULTCOLLFRAMES is shown in [Table 9-484](#).

Return to the [Summary Table](#).

Total number of transmitted frames experiencing multiple collisions

**Table 9-484. CPSW\_NC\_STAT\_0\_TXMULTCOLLFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames experiencing multiple collisions

### 9.6.453 CPSW\_NC\_STAT\_0\_TXOCTETS Register (Offset = 0003A064h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TXOCTETS is shown in [Table 9-485](#).

Return to the [Summary Table](#).

Total number of bytes in all good frames transmitted

**Table 9-485. CPSW\_NC\_STAT\_0\_TXOCTETS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of bytes in all good frames transmitted

### 9.6.454 CPSW\_NC\_STAT\_0\_OCTETFRAMES64 Register (Offset = 0003A068h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_OCTETFRAMES64 is shown in [Table 9-486](#).

Return to the [Summary Table](#).

Total number of 64-byte frames received and transmitted

**Table 9-486. CPSW\_NC\_STAT\_0\_OCTETFRAMES64 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of 64-byte frames received and transmitted

### 9.6.455 CPSW\_NC\_STAT\_0\_OCTETFRAMES65T127 Register (Offset = 0003A06Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_OCTETFRAMES65T127 is shown in [Table 9-487](#).

Return to the [Summary Table](#).

Total number of frames of size 65 to 127 bytes received and transmitted

**Table 9-487. CPSW\_NC\_STAT\_0\_OCTETFRAMES65T127 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 65 to 127 bytes received and transmitted

### 9.6.456 CPSW\_NC\_STAT\_0\_OCTETFRAMES128T255 Register (Offset = 0003A070h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_OCTETFRAMES128T255 is shown in [Table 9-488](#).

Return to the [Summary Table](#).

Total number of frames of size 128 to 255 bytes received and transmitted

**Table 9-488. CPSW\_NC\_STAT\_0\_OCTETFRAMES128T255 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 128 to 255 bytes received and transmitted

### 9.6.457 CPSW\_NC\_STAT\_0\_OCTETFRAMES256T511 Register (Offset = 0003A074h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_OCTETFRAMES256T511 is shown in [Table 9-489](#).

Return to the [Summary Table](#).

Total number of frames of size 256 to 511 bytes received and transmitted

**Table 9-489. CPSW\_NC\_STAT\_0\_OCTETFRAMES256T511 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 256 to 511 bytes received and transmitted

### 9.6.458 CPSW\_NC\_STAT\_0\_OCTETFRAMES512T1023 Register (Offset = 0003A078h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_OCTETFRAMES512T1023 is shown in [Table 9-490](#).

Return to the [Summary Table](#).

Total number of frames of size 512 to 1023 bytes received and transmitted

**Table 9-490. CPSW\_NC\_STAT\_0\_OCTETFRAMES512T1023 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 512 to 1023 bytes received and transmitted



### 9.6.459 CPSW\_NC\_STAT\_0\_OCTETFRAMES1024TUP Register (Offset = 0003A07Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_OCTETFRAMES1024TUP is shown in [Table 9-491](#).

Return to the [Summary Table](#).

Total number of frames of size 1024 to rx\_maxlen bytes received and 1024 bytes or greater transmitted

**Table 9-491. CPSW\_NC\_STAT\_0\_OCTETFRAMES1024TUP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 1024 to rx_maxlen bytes received and 1024 bytes or greater transmitted

### 9.6.460 CPSW\_NC\_STAT\_0\_NETOCTETS Register (Offset = 0003A080h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_NETOCTETS is shown in [Table 9-492](#).

Return to the [Summary Table](#).

Total number of bytes received and transmitted

**Table 9-492. CPSW\_NC\_STAT\_0\_NETOCTETS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of bytes received and transmitted

### 9.6.461 CPSW\_NC\_STAT\_0\_RX\_BOTTOM\_OF\_FIFO\_DROP Register (Offset = 0003A084h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RX\_BOTTOM\_OF\_FIFO\_DROP is shown in [Table 9-493](#).

Return to the [Summary Table](#).

Receive Bottom of FIFO Drop

**Table 9-493. CPSW\_NC\_STAT\_0\_RX\_BOTTOM\_OF\_FIFO\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Receive Bottom of FIFO Drop

### 9.6.462 CPSW\_NC\_STAT\_0\_PORTMASK\_DROP Register (Offset = 0003A088h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_PORTMASK\_DROP is shown in [Table 9-494](#).

Return to the [Summary Table](#).

Total number of dropped frames received due to portmask

**Table 9-494. CPSW\_NC\_STAT\_0\_PORTMASK\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames received due to portmask

### 9.6.463 CPSW\_NC\_STAT\_0\_RX\_TOP\_OF\_FIFO\_DROP Register (Offset = 0003A08Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_RX\_TOP\_OF\_FIFO\_DROP is shown in [Table 9-495](#).

Return to the [Summary Table](#).

Receive Top of FIFO Drop

**Table 9-495. CPSW\_NC\_STAT\_0\_RX\_TOP\_OF\_FIFO\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Receive Top of FIFO Drop

### 9.6.464 CPSW\_NC\_STAT\_0\_ALE\_RATE\_LIMIT\_DROP Register (Offset = 0003A090h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_RATE\_LIMIT\_DROP is shown in [Table 9-496](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Rate Limiting

**Table 9-496. CPSW\_NC\_STAT\_0\_ALE\_RATE\_LIMIT\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Rate Limiting

### 9.6.465 CPSW\_NC\_STAT\_0\_ALE\_VID\_INGRESS\_DROP Register (Offset = 0003A094h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_VID\_INGRESS\_DROP is shown in [Table 9-497](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE VID Ingress

**Table 9-497. CPSW\_NC\_STAT\_0\_ALE\_VID\_INGRESS\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE VID Ingress

### 9.6.466 CPSW\_NC\_STAT\_0\_ALE\_DA\_EQ\_SA\_DROP Register (Offset = 0003A098h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_DA\_EQ\_SA\_DROP is shown in [Table 9-498](#).

Return to the [Summary Table](#).

Total number of dropped frames due to DA=SA

**Table 9-498. CPSW\_NC\_STAT\_0\_ALE\_DA\_EQ\_SA\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to DA=SA



### 9.6.467 CPSW\_NC\_STAT\_0\_ALE\_BLOCK\_DROP Register (Offset = 0003A09Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_BLOCK\_DROP is shown in [Table 9-499](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Block Mode

**Table 9-499. CPSW\_NC\_STAT\_0\_ALE\_BLOCK\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Block Mode

### 9.6.468 CPSW\_NC\_STAT\_0\_ALE\_SECURE\_DROP Register (Offset = 0003A0A0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_SECURE\_DROP is shown in [Table 9-500](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Secure Mode

**Table 9-500. CPSW\_NC\_STAT\_0\_ALE\_SECURE\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Secure Mode

### 9.6.469 CPSW\_NC\_STAT\_0\_ALE\_AUTH\_DROP Register (Offset = 0003A0A4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_AUTH\_DROP is shown in [Table 9-501](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Authentication

**Table 9-501. CPSW\_NC\_STAT\_0\_ALE\_AUTH\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Authentication

### 9.6.470 CPSW\_NC\_STAT\_0\_ALE\_UNKN\_UNI Register (Offset = 0003A0A8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_UNKN\_UNI is shown in [Table 9-502](#).

Return to the [Summary Table](#).

ALE Receive Unknown Unicast

**Table 9-502. CPSW\_NC\_STAT\_0\_ALE\_UNKN\_UNI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Unicast

### 9.6.471 CPSW\_NC\_STAT\_0\_ALE\_UNKN\_UNI\_BCNT Register (Offset = 0003A0ACh) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_UNKN\_UNI\_BCNT is shown in [Table 9-503](#).

Return to the [Summary Table](#).

ALE Receive Unknown Unicast Bytecount

**Table 9-503. CPSW\_NC\_STAT\_0\_ALE\_UNKN\_UNI\_BCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Unicast Bytecount

### 9.6.472 CPSW\_NC\_STAT\_0\_ALE\_UNKN\_MLT Register (Offset = 0003A0B0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_UNKN\_MLT is shown in [Table 9-504](#).

Return to the [Summary Table](#).

ALE Receive Unknown Multicast

**Table 9-504. CPSW\_NC\_STAT\_0\_ALE\_UNKN\_MLT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Multicast

### 9.6.473 CPSW\_NC\_STAT\_0\_ALE\_UNKN\_MLT\_BCNT Register (Offset = 0003A0B4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_UNKN\_MLT\_BCNT is shown in [Table 9-505](#).

Return to the [Summary Table](#).

ALE Receive Unknown Multicast Bytecount

**Table 9-505. CPSW\_NC\_STAT\_0\_ALE\_UNKN\_MLT\_BCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Multicast Bytecount

### 9.6.474 CPSW\_NC\_STAT\_0\_ALE\_UNKN\_BRD Register (Offset = 0003A0B8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_UNKN\_BRD is shown in [Table 9-506](#).

Return to the [Summary Table](#).

ALE Receive Unknown Broadcast

**Table 9-506. CPSW\_NC\_STAT\_0\_ALE\_UNKN\_BRD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Broadcast



### 9.6.475 CPSW\_NC\_STAT\_0\_ALE\_UNKN\_BRD\_BCNT Register (Offset = 0003A0BCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_UNKN\_BRD\_BCNT is shown in [Table 9-507](#).

Return to the [Summary Table](#).

ALE Receive Unknown Broadcast Bytecount

**Table 9-507. CPSW\_NC\_STAT\_0\_ALE\_UNKN\_BRD\_BCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Broadcast Bytecount

### 9.6.476 CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH Register (Offset = 0003A0C0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH is shown in [Table 9-508](#).

Return to the [Summary Table](#).

ALE Policer Matched

**Table 9-508. CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Policer Matched

### 9.6.477 CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH\_RED Register (Offset = 0003A0C4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH\_RED is shown in [Table 9-509](#).

Return to the [Summary Table](#).

ALE Policer Matched and Condition Red

**Table 9-509. CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH\_RED Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Policer Matched and Condition Red

### 9.6.478 CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH\_YELLOW Register (Offset = 0003A0C8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH\_YELLOW is shown in [Table 9-510](#).

Return to the [Summary Table](#).

ALE Policer Matched and Condition Yellow

**Table 9-510. CPSW\_NC\_STAT\_0\_ALE\_POL\_MATCH\_YELLOW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Policer Matched and Condition Yellow

### 9.6.479 CPSW\_NC\_STAT\_0\_ALE\_MULT\_SA\_DROP Register (Offset = 0003A0CCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_MULT\_SA\_DROP is shown in [Table 9-511](#).

Return to the [Summary Table](#).

ALE Multicast Source Address Drop

**Table 9-511. CPSW\_NC\_STAT\_0\_ALE\_MULT\_SA\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Multicast Source Address drop

### 9.6.480 CPSW\_NC\_STAT\_0\_ALE\_DUAL\_VLAN\_DROP Register (Offset = 0003A0D0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_DUAL\_VLAN\_DROP is shown in [Table 9-512](#).

Return to the [Summary Table](#).

ALE Dual VLAN Drop

**Table 9-512. CPSW\_NC\_STAT\_0\_ALE\_DUAL\_VLAN\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Dual VLAN drop

### 9.6.481 CPSW\_NC\_STAT\_0\_ALE\_LEN\_ERROR\_DROP Register (Offset = 0003A0D4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_LEN\_ERROR\_DROP is shown in [Table 9-513](#).

Return to the [Summary Table](#).

ALE Length Error Drop

**Table 9-513. CPSW\_NC\_STAT\_0\_ALE\_LEN\_ERROR\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Length Error drop

### 9.6.482 CPSW\_NC\_STAT\_0\_ALE\_IP\_NEXT\_HDR\_DROP Register (Offset = 0003A0D8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_IP\_NEXT\_HDR\_DROP is shown in [Table 9-514](#).

Return to the [Summary Table](#).

ALE IP Next Header Drop

**Table 9-514. CPSW\_NC\_STAT\_0\_ALE\_IP\_NEXT\_HDR\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Next Header drop



### 9.6.483 CPSW\_NC\_STAT\_0\_ALE\_IPV4\_FRAG\_DROP Register (Offset = 0003A0DCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_ALE\_IPV4\_FRAG\_DROP is shown in [Table 9-515](#).

Return to the [Summary Table](#).

ALE IPV4 Frag Drop

**Table 9-515. CPSW\_NC\_STAT\_0\_ALE\_IPV4\_FRAG\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE IPV4 Fragment drop

### 9.6.484 CPSW\_NC\_STAT\_0\_TX\_MEMORY\_PROTECT\_ERROR Register (Offset = 0003A17Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_0\_TX\_MEMORY\_PROTECT\_ERROR is shown in [Table 9-516](#).

Return to the [Summary Table](#).

Transmit Memory Protect CRC Error

**Table 9-516. CPSW\_NC\_STAT\_0\_TX\_MEMORY\_PROTECT\_ERROR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	COUNT	R/W	0h	Transmit Memory Protect CRC Error

### 9.6.485 CPSW\_NC\_STAT\_1\_RXGOODFRAMES Register (Offset = 0003A200h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXGOODFRAMES is shown in [Table 9-517](#).

Return to the [Summary Table](#).

Total number of good frames received

**Table 9-517. CPSW\_NC\_STAT\_1\_RXGOODFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good frames received

### 9.6.486 CPSW\_NC\_STAT\_1\_RXBROADCASTFRAMES Register (Offset = 0003A204h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXBROADCASTFRAMES is shown in [Table 9-518](#).

Return to the [Summary Table](#).

Total number of good broadcast frames received

**Table 9-518. CPSW\_NC\_STAT\_1\_RXBROADCASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good broadcast frames received

### 9.6.487 CPSW\_NC\_STAT\_1\_RXMULTICASTFRAMES Register (Offset = 0003A208h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXMULTICASTFRAMES is shown in [Table 9-519](#).

Return to the [Summary Table](#).

Total number of good multicast frames received

**Table 9-519. CPSW\_NC\_STAT\_1\_RXMULTICASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good multicast frames received

### 9.6.488 CPSW\_NC\_STAT\_1\_RXPAUSEFRAMES Register (Offset = 0003A20Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXPAUSEFRAMES is shown in [Table 9-520](#).

Return to the [Summary Table](#).

Total number of pause frames received

**Table 9-520. CPSW\_NC\_STAT\_1\_RXPAUSEFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of pause frames received

### 9.6.489 CPSW\_NC\_STAT\_1\_RXCRCERRORS Register (Offset = 0003A210h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXCRCERRORS is shown in [Table 9-521](#).

Return to the [Summary Table](#).

Total number of CRC errors frames received

**Table 9-521. CPSW\_NC\_STAT\_1\_RXCRCERRORS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of CRC errors frames received

### 9.6.490 CPSW\_NC\_STAT\_1\_RXALIGNCODEERRORS Register (Offset = 0003A214h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXALIGNCODEERRORS is shown in [Table 9-522](#).

Return to the [Summary Table](#).

Total number of alignment/code errors received

**Table 9-522. CPSW\_NC\_STAT\_1\_RXALIGNCODEERRORS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of alignment/code errors received



### 9.6.491 CPSW\_NC\_STAT\_1\_RXOVERSIZEDFRAMES Register (Offset = 0003A218h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXOVERSIZEDFRAMES is shown in [Table 9-523](#).

Return to the [Summary Table](#).

Total number of oversized frames received

**Table 9-523. CPSW\_NC\_STAT\_1\_RXOVERSIZEDFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of oversized frames received

### 9.6.492 CPSW\_NC\_STAT\_1\_RXJABBERFRAMES Register (Offset = 0003A21Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXJABBERFRAMES is shown in [Table 9-524](#).

Return to the [Summary Table](#).

Total number of jabber frames received

**Table 9-524. CPSW\_NC\_STAT\_1\_RXJABBERFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of jabber frames received

### 9.6.493 CPSW\_NC\_STAT\_1\_RXUNDERSIZEDFRAMES Register (Offset = 0003A220h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXUNDERSIZEDFRAMES is shown in [Table 9-525](#).

Return to the [Summary Table](#).

Total number of undersized frames received

**Table 9-525. CPSW\_NC\_STAT\_1\_RXUNDERSIZEDFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of undersized frames received

### 9.6.494 CPSW\_NC\_STAT\_1\_RXFRAGMENTS Register (Offset = 0003A224h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXFRAGMENTS is shown in [Table 9-526](#).

Return to the [Summary Table](#).

Total number of fragmented frames received

**Table 9-526. CPSW\_NC\_STAT\_1\_RXFRAGMENTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of fragmented frames received

### 9.6.495 CPSW\_NC\_STAT\_1\_ALE\_DROP Register (Offset = 0003A228h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_DROP is shown in [Table 9-527](#).

Return to the [Summary Table](#).

Total number of frames dropped by the ALE

**Table 9-527. CPSW\_NC\_STAT\_1\_ALE\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames dropped by the ALE

### 9.6.496 CPSW\_NC\_STAT\_1\_ALE\_OVERRUN\_DROP Register (Offset = 0003A22Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_OVERRUN\_DROP is shown in [Table 9-528](#).

Return to the [Summary Table](#).

Total number of overrun frames dropped by the ALE

**Table 9-528. CPSW\_NC\_STAT\_1\_ALE\_OVERRUN\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of overrun frames dropped by the ALE

### 9.6.497 CPSW\_NC\_STAT\_1\_RXOCTETS Register (Offset = 0003A230h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXOCTETS is shown in [Table 9-529](#).

Return to the [Summary Table](#).

Total number of received bytes in good frames

**Table 9-529. CPSW\_NC\_STAT\_1\_RXOCTETS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of received bytes in good frames

### 9.6.498 CPSW\_NC\_STAT\_1\_TXGOODFRAMES Register (Offset = 0003A234h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXGOODFRAMES is shown in [Table 9-530](#).

Return to the [Summary Table](#).

Total number of good frames transmitted

**Table 9-530. CPSW\_NC\_STAT\_1\_TXGOODFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good frames transmitted



### 9.6.499 CPSW\_NC\_STAT\_1\_TXBROADCASTFRAMES Register (Offset = 0003A238h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXBROADCASTFRAMES is shown in [Table 9-531](#).

Return to the [Summary Table](#).

Total number of good broadcast frames transmitted

**Table 9-531. CPSW\_NC\_STAT\_1\_TXBROADCASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good broadcast frames transmitted

### 9.6.500 CPSW\_NC\_STAT\_1\_TXMULTICASTFRAMES Register (Offset = 0003A23Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXMULTICASTFRAMES is shown in [Table 9-532](#).

Return to the [Summary Table](#).

Total number of good multicast frames transmitted

**Table 9-532. CPSW\_NC\_STAT\_1\_TXMULTICASTFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of good multicast frames transmitted

### 9.6.501 CPSW\_NC\_STAT\_1\_TXPAUSEFRAMES Register (Offset = 0003A240h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXPAUSEFRAMES is shown in [Table 9-533](#).

Return to the [Summary Table](#).

Total number of pause frames transmitted

**Table 9-533. CPSW\_NC\_STAT\_1\_TXPAUSEFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of pause frames transmitted

### 9.6.502 CPSW\_NC\_STAT\_1\_TXDEFERREDFRAMES Register (Offset = 0003A244h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXDEFERREDFRAMES is shown in [Table 9-534](#).

Return to the [Summary Table](#).

Total number of deferred frames transmitted

**Table 9-534. CPSW\_NC\_STAT\_1\_TXDEFERREDFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of deferred frames transmitted

### 9.6.503 CPSW\_NC\_STAT\_1\_TXCOLLISIONFRAMES Register (Offset = 0003A248h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXCOLLISIONFRAMES is shown in [Table 9-535](#).

Return to the [Summary Table](#).

Total number of transmitted frames experiencing a collision

**Table 9-535. CPSW\_NC\_STAT\_1\_TXCOLLISIONFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames experiencing a collision

### 9.6.504 CPSW\_NC\_STAT\_1\_TXSINGLECOLLFRAMES Register (Offset = 0003A24Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXSINGLECOLLFRAMES is shown in [Table 9-536](#).

Return to the [Summary Table](#).

Total number of transmitted frames experiencing a single collision

**Table 9-536. CPSW\_NC\_STAT\_1\_TXSINGLECOLLFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames experiencing a single collision

### 9.6.505 CPSW\_NC\_STAT\_1\_TXMULTCOLLFRAMES Register (Offset = 0003A250h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXMULTCOLLFRAMES is shown in [Table 9-537](#).

Return to the [Summary Table](#).

Total number of transmitted frames experiencing multiple collisions

**Table 9-537. CPSW\_NC\_STAT\_1\_TXMULTCOLLFRAMES Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames experiencing multiple collisions

### 9.6.506 CPSW\_NC\_STAT\_1\_TXEXCESSIVECOLLISIONS Register (Offset = 0003A254h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXEXCESSIVECOLLISIONS is shown in [Table 9-538](#).

Return to the [Summary Table](#).

Total number of transmitted frames abandoned due to excessive collisions

**Table 9-538. CPSW\_NC\_STAT\_1\_TXEXCESSIVECOLLISIONS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames abandoned due to excessive collisions



### 9.6.507 CPSW\_NC\_STAT\_1\_TXLATECOLLISIONS Register (Offset = 0003A258h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXLATECOLLISIONS is shown in [Table 9-539](#).

Return to the [Summary Table](#).

Total number of transmitted frames abandoned due to a late collision

**Table 9-539. CPSW\_NC\_STAT\_1\_TXLATECOLLISIONS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames abandoned due to a late collision

### 9.6.508 CPSW\_NC\_STAT\_1\_RXIPGERROR Register (Offset = 0003A25Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RXIPGERROR is shown in [Table 9-540](#).

Return to the [Summary Table](#).

Total number of receive inter-packet gap errors (10G only)

**Table 9-540. CPSW\_NC\_STAT\_1\_RXIPGERROR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of receive inter-packet gap errors (10G only)

### 9.6.509 CPSW\_NC\_STAT\_1\_TXCARRIERSENSEERRORS Register (Offset = 0003A260h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXCARRIERSENSEERRORS is shown in [Table 9-541](#).

Return to the [Summary Table](#).

Total number of transmitted frames that experienced a carrier loss

**Table 9-541. CPSW\_NC\_STAT\_1\_TXCARRIERSENSEERRORS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of transmitted frames that experienced a carrier loss

### 9.6.510 CPSW\_NC\_STAT\_1\_TXOCTETS Register (Offset = 0003A264h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TXOCTETS is shown in [Table 9-542](#).

Return to the [Summary Table](#).

Total number of bytes in all good frames transmitted

**Table 9-542. CPSW\_NC\_STAT\_1\_TXOCTETS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of bytes in all good frames transmitted

### 9.6.511 CPSW\_NC\_STAT\_1\_OCTETFRAMES64 Register (Offset = 0003A268h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_OCTETFRAMES64 is shown in [Table 9-543](#).

Return to the [Summary Table](#).

Total number of 64-byte frames received and transmitted

**Table 9-543. CPSW\_NC\_STAT\_1\_OCTETFRAMES64 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of 64-byte frames received and transmitted

### 9.6.512 CPSW\_NC\_STAT\_1\_OCTETFRAMES65T127 Register (Offset = 0003A26Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_OCTETFRAMES65T127 is shown in [Table 9-544](#).

Return to the [Summary Table](#).

Total number of frames of size 65 to 127 bytes received and transmitted

**Table 9-544. CPSW\_NC\_STAT\_1\_OCTETFRAMES65T127 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 65 to 127 bytes received and transmitted

### 9.6.513 CPSW\_NC\_STAT\_1\_OCTETFRAMES128T255 Register (Offset = 0003A270h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_OCTETFRAMES128T255 is shown in [Table 9-545](#).

Return to the [Summary Table](#).

Total number of frames of size 128 to 255 bytes received and transmitted

**Table 9-545. CPSW\_NC\_STAT\_1\_OCTETFRAMES128T255 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 128 to 255 bytes received and transmitted

### 9.6.514 CPSW\_NC\_STAT\_1\_OCTETFRAMES256T511 Register (Offset = 0003A274h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_OCTETFRAMES256T511 is shown in [Table 9-546](#).

Return to the [Summary Table](#).

Total number of frames of size 256 to 511 bytes received and transmitted

**Table 9-546. CPSW\_NC\_STAT\_1\_OCTETFRAMES256T511 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 256 to 511 bytes received and transmitted



### 9.6.515 CPSW\_NC\_STAT\_1\_OCTETFRAMES512T1023 Register (Offset = 0003A278h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_OCTETFRAMES512T1023 is shown in [Table 9-547](#).

Return to the [Summary Table](#).

Total number of frames of size 512 to 1023 bytes received and transmitted

**Table 9-547. CPSW\_NC\_STAT\_1\_OCTETFRAMES512T1023 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 512 to 1023 bytes received and transmitted

### 9.6.516 CPSW\_NC\_STAT\_1\_OCTETFRAMES1024TUP Register (Offset = 0003A27Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_OCTETFRAMES1024TUP is shown in [Table 9-548](#).

Return to the [Summary Table](#).

Total number of frames of size 1024 to rx\_maxlen bytes received and 1024 bytes or greater transmitted

**Table 9-548. CPSW\_NC\_STAT\_1\_OCTETFRAMES1024TUP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of frames of size 1024 to rx_maxlen bytes received and 1024 bytes or greater transmitted

### 9.6.517 CPSW\_NC\_STAT\_1\_NETOCTETS Register (Offset = 0003A280h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_NETOCTETS is shown in [Table 9-549](#).

Return to the [Summary Table](#).

Total number of bytes received and transmitted

**Table 9-549. CPSW\_NC\_STAT\_1\_NETOCTETS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of bytes received and transmitted

### 9.6.518 CPSW\_NC\_STAT\_1\_RX\_BOTTOM\_OF\_FIFO\_DROP Register (Offset = 0003A284h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RX\_BOTTOM\_OF\_FIFO\_DROP is shown in [Table 9-550](#).

Return to the [Summary Table](#).

Receive Bottom of FIFO Drop

**Table 9-550. CPSW\_NC\_STAT\_1\_RX\_BOTTOM\_OF\_FIFO\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Receive Bottom of FIFO Drop

### 9.6.519 CPSW\_NC\_STAT\_1\_PORTMASK\_DROP Register (Offset = 0003A288h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_PORTMASK\_DROP is shown in [Table 9-551](#).

Return to the [Summary Table](#).

Total number of dropped frames received due to portmask

**Table 9-551. CPSW\_NC\_STAT\_1\_PORTMASK\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames received due to portmask

### 9.6.520 CPSW\_NC\_STAT\_1\_RX\_TOP\_OF\_FIFO\_DROP Register (Offset = 0003A28Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_RX\_TOP\_OF\_FIFO\_DROP is shown in [Table 9-552](#).

Return to the [Summary Table](#).

Receive Top of FIFO Drop

**Table 9-552. CPSW\_NC\_STAT\_1\_RX\_TOP\_OF\_FIFO\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Receive Top of FIFO Drop

### 9.6.521 CPSW\_NC\_STAT\_1\_ALE\_RATE\_LIMIT\_DROP Register (Offset = 0003A290h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_RATE\_LIMIT\_DROP is shown in [Table 9-553](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Rate Limiting

**Table 9-553. CPSW\_NC\_STAT\_1\_ALE\_RATE\_LIMIT\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Rate Limiting

### 9.6.522 CPSW\_NC\_STAT\_1\_ALE\_VID\_INGRESS\_DROP Register (Offset = 0003A294h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_VID\_INGRESS\_DROP is shown in [Table 9-554](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE VID Ingress

**Table 9-554. CPSW\_NC\_STAT\_1\_ALE\_VID\_INGRESS\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE VID Ingress



### 9.6.523 CPSW\_NC\_STAT\_1\_ALE\_DA\_EQ\_SA\_DROP Register (Offset = 0003A298h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_DA\_EQ\_SA\_DROP is shown in [Table 9-555](#).

Return to the [Summary Table](#).

Total number of dropped frames due to DA=SA

**Table 9-555. CPSW\_NC\_STAT\_1\_ALE\_DA\_EQ\_SA\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to DA=SA

### 9.6.524 CPSW\_NC\_STAT\_1\_ALE\_BLOCK\_DROP Register (Offset = 0003A29Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_BLOCK\_DROP is shown in [Table 9-556](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Block Mode

**Table 9-556. CPSW\_NC\_STAT\_1\_ALE\_BLOCK\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Block Mode

### 9.6.525 CPSW\_NC\_STAT\_1\_ALE\_SECURE\_DROP Register (Offset = 0003A2A0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_SECURE\_DROP is shown in [Table 9-557](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Secure Mode

**Table 9-557. CPSW\_NC\_STAT\_1\_ALE\_SECURE\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Secure Mode

### 9.6.526 CPSW\_NC\_STAT\_1\_ALE\_AUTH\_DROP Register (Offset = 0003A2A4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_AUTH\_DROP is shown in [Table 9-558](#).

Return to the [Summary Table](#).

Total number of dropped frames due to ALE Authentication

**Table 9-558. CPSW\_NC\_STAT\_1\_ALE\_AUTH\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	Total number of dropped frames due to ALE Authentication

### 9.6.527 CPSW\_NC\_STAT\_1\_ALE\_UNKN\_UNI Register (Offset = 0003A2A8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_UNKN\_UNI is shown in [Table 9-559](#).

Return to the [Summary Table](#).

ALE Receive Unknown Unicast

**Table 9-559. CPSW\_NC\_STAT\_1\_ALE\_UNKN\_UNI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Unicast

### 9.6.528 CPSW\_NC\_STAT\_1\_ALE\_UNKN\_UNI\_BCNT Register (Offset = 0003A2ACh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_UNKN\_UNI\_BCNT is shown in [Table 9-560](#).

Return to the [Summary Table](#).

ALE Receive Unknown Unicast Bytecount

**Table 9-560. CPSW\_NC\_STAT\_1\_ALE\_UNKN\_UNI\_BCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Unicast Bytecount

### 9.6.529 CPSW\_NC\_STAT\_1\_ALE\_UNKN\_MLT Register (Offset = 0003A2B0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_UNKN\_MLT is shown in [Table 9-561](#).

Return to the [Summary Table](#).

ALE Receive Unknown Multicast

**Table 9-561. CPSW\_NC\_STAT\_1\_ALE\_UNKN\_MLT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Multicast

### 9.6.530 CPSW\_NC\_STAT\_1\_ALE\_UNKN\_MLT\_BCNT Register (Offset = 0003A2B4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_UNKN\_MLT\_BCNT is shown in [Table 9-562](#).

Return to the [Summary Table](#).

ALE Receive Unknown Multicast Bytecount

**Table 9-562. CPSW\_NC\_STAT\_1\_ALE\_UNKN\_MLT\_BCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Multicast Bytecount



### 9.6.531 CPSW\_NC\_STAT\_1\_ALE\_UNKN\_BRD Register (Offset = 0003A2B8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_UNKN\_BRD is shown in [Table 9-563](#).

Return to the [Summary Table](#).

ALE Receive Unknown Broadcast

**Table 9-563. CPSW\_NC\_STAT\_1\_ALE\_UNKN\_BRD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Broadcast

### 9.6.532 CPSW\_NC\_STAT\_1\_ALE\_UNKN\_BRD\_BCNT Register (Offset = 0003A2BCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_UNKN\_BRD\_BCNT is shown in [Table 9-564](#).

Return to the [Summary Table](#).

ALE Receive Unknown Broadcast Bytecount

**Table 9-564. CPSW\_NC\_STAT\_1\_ALE\_UNKN\_BRD\_BCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Receive Unknown Broadcast Bytecount

### 9.6.533 CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH Register (Offset = 0003A2C0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH is shown in [Table 9-565](#).

Return to the [Summary Table](#).

ALE Policer Matched

**Table 9-565. CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Policer Matched

### 9.6.534 CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH\_RED Register (Offset = 0003A2C4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH\_RED is shown in [Table 9-566](#).

Return to the [Summary Table](#).

ALE Policer Matched and Condition Red

**Table 9-566. CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH\_RED Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Policer Matched and Condition Red

### 9.6.535 CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH\_YELLOW Register (Offset = 0003A2C8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH\_YELLOW is shown in [Table 9-567](#).

Return to the [Summary Table](#).

ALE Policer Matched and Condition Yellow

**Table 9-567. CPSW\_NC\_STAT\_1\_ALE\_POL\_MATCH\_YELLOW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Policer Matched and Condition Yellow

### 9.6.536 CPSW\_NC\_STAT\_1\_ALE\_MULT\_SA\_DROP Register (Offset = 0003A2CCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_MULT\_SA\_DROP is shown in [Table 9-568](#).

Return to the [Summary Table](#).

ALE Multicast Source Address Drop

**Table 9-568. CPSW\_NC\_STAT\_1\_ALE\_MULT\_SA\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Multicast Source Address drop

### 9.6.537 CPSW\_NC\_STAT\_1\_ALE\_DUAL\_VLAN\_DROP Register (Offset = 0003A2D0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_DUAL\_VLAN\_DROP is shown in [Table 9-569](#).

Return to the [Summary Table](#).

ALE Dual VLAN Drop

**Table 9-569. CPSW\_NC\_STAT\_1\_ALE\_DUAL\_VLAN\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Dual VLAN drop

### 9.6.538 CPSW\_NC\_STAT\_1\_ALE\_LEN\_ERROR\_DROP Register (Offset = 0003A2D4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_LEN\_ERROR\_DROP is shown in [Table 9-570](#).

Return to the [Summary Table](#).

ALE Length Error Drop

**Table 9-570. CPSW\_NC\_STAT\_1\_ALE\_LEN\_ERROR\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Length Error drop



### 9.6.539 CPSW\_NC\_STAT\_1\_ALE\_IP\_NEXT\_HDR\_DROP Register (Offset = 0003A2D8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_IP\_NEXT\_HDR\_DROP is shown in [Table 9-571](#).

Return to the [Summary Table](#).

ALE IP Next Header Drop

**Table 9-571. CPSW\_NC\_STAT\_1\_ALE\_IP\_NEXT\_HDR\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE Next Header drop

### 9.6.540 CPSW\_NC\_STAT\_1\_ALE\_IPV4\_FRAG\_DROP Register (Offset = 0003A2DCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ALE\_IPV4\_FRAG\_DROP is shown in [Table 9-572](#).

Return to the [Summary Table](#).

ALE IPV4 Frag Drop

**Table 9-572. CPSW\_NC\_STAT\_1\_ALE\_IPV4\_FRAG\_DROP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COUNT	R/W	0h	ALE IPV4 Fragment drop

### 9.6.541 CPSW\_NC\_STAT\_1\_TX\_MEMORY\_PROTECT\_ERROR Register (Offset = 0003A37Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_TX\_MEMORY\_PROTECT\_ERROR is shown in [Table 9-573](#).

Return to the [Summary Table](#).

Transmit Memory Protect CRC Error

**Table 9-573. CPSW\_NC\_STAT\_1\_TX\_MEMORY\_PROTECT\_ERROR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	COUNT	R/W	0h	Transmit Memory Protect CRC Error

### 9.6.542 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_0 Register (Offset = 0003A380h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_0 is shown in [Table 9-574](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-574. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.543 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_1 Register (Offset = 0003A384h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_1 is shown in [Table 9-575](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-575. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.544 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_2 Register (Offset = 0003A388h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_2 is shown in [Table 9-576](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-576. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.545 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_3 Register (Offset = 0003A38Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_3 is shown in [Table 9-577](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-577. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.546 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_4 Register (Offset = 0003A390h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_4 is shown in [Table 9-578](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-578. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count



### 9.6.547 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_5 Register (Offset = 0003A394h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_5 is shown in [Table 9-579](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-579. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.548 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_6 Register (Offset = 0003A398h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_6 is shown in [Table 9-580](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-580. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.549 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_7 Register (Offset = 0003A39Ch) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_7 is shown in [Table 9-581](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Count

**Table 9-581. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN	R/W	0h	ENET TX Priority Packet Count

### 9.6.550 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_0 Register (Offset = 0003A3A0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_0 is shown in [Table 9-582](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-582. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.551 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_1 Register (Offset = 0003A3A4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_1 is shown in [Table 9-583](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-583. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.552 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_2 Register (Offset = 0003A3A8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_2 is shown in [Table 9-584](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-584. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.553 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_3 Register (Offset = 0003A3ACh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_3 is shown in [Table 9-585](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-585. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.554 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_4 Register (Offset = 0003A3B0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_4 is shown in [Table 9-586](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-586. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count



### 9.6.555 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_5 Register (Offset = 0003A3B4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_5 is shown in [Table 9-587](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-587. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.556 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_6 Register (Offset = 0003A3B8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_6 is shown in [Table 9-588](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-588. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.557 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_7 Register (Offset = 0003A3BCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_7 is shown in [Table 9-589](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Byte Count

**Table 9-589. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_BCNT\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Byte Count

### 9.6.558 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_0 Register (Offset = 0003A3C0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_0 is shown in [Table 9-590](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-590. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.559 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_1 Register (Offset = 0003A3C4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_1 is shown in [Table 9-591](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-591. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.560 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_2 Register (Offset = 0003A3C8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_2 is shown in [Table 9-592](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-592. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.561 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_3 Register (Offset = 0003A3CCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_3 is shown in [Table 9-593](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-593. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.562 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_4 Register (Offset = 0003A3D0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_4 is shown in [Table 9-594](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-594. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count



### 9.6.563 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_5 Register (Offset = 0003A3D4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_5 is shown in [Table 9-595](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-595. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.564 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_6 Register (Offset = 0003A3D8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_6 is shown in [Table 9-596](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-596. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.565 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_7 Register (Offset = 0003A3DCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_7 is shown in [Table 9-597](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Count

**Table 9-597. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP	R/W	0h	ENET Port n PRIORITY N Packet Drop Count

### 9.6.566 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_0 Register (Offset = 0003A3E0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_0 is shown in [Table 9-598](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-598. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.567 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_1 Register (Offset = 0003A3E4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_1 is shown in [Table 9-599](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-599. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.568 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_2 Register (Offset = 0003A3E8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_2 is shown in [Table 9-600](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-600. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.569 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_3 Register (Offset = 0003A3ECh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_3 is shown in [Table 9-601](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-601. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.570 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_4 Register (Offset = 0003A3F0h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_4 is shown in [Table 9-602](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-602. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count



### 9.6.571 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_5 Register (Offset = 0003A3F4h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_5 is shown in [Table 9-603](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-603. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.572 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_6 Register (Offset = 0003A3F8h) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_6 is shown in [Table 9-604](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-604. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.573 CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_7 Register (Offset = 0003A3FCh) [Reset = 00000000h]

CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_7 is shown in [Table 9-605](#).

Return to the [Summary Table](#).

ENET Port n PRIORITY N Packet Drop Byte Count

**Table 9-605. CPSW\_NC\_STAT\_1\_ENET\_PN\_TX\_PRI\_DROP\_BCNT\_REG\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PN_TX_PRIN_DROP_BCNT	R/W	0h	ENET Port n PRIORITY N Packet Drop Byte Count

### 9.6.574 IDVER\_REG Register (Offset = 0003D000h) [Reset = 4E8A010Dh]

IDVER\_REG is shown in [Table 9-606](#).

Return to the [Summary Table](#).

Identification and Version Register

**Table 9-606. IDVER\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	TX_IDENT	R	4E8Ah	Identification value
15-11	RTL_VER	R	0h	RTL version value
10-8	MAJOR_VER	R	1h	Major version value
7-0	MINOR_VER	R	Dh	Minor version value

### 9.6.575 CPTS\_CONTROL\_REG Register (Offset = 0003D004h) [Reset = 0XXX0004h]

CPTS\_CONTROL\_REG is shown in [Table 9-607](#).

Return to the [Summary Table](#).

Time Sync Control Register

**Table 9-607. CPTS\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	TS_SYNC_SEL	R/W	0h	TS_SYNC output timestamp counter bit select
27-18	RESERVED	R	0h	
17	TS_GENF_CLR_EN	R/W	0h	Enable for GENF clear when length is zero
16	TS_RX_NO_EVENT	R/W	0h	Receive Produces no Events
15	HW8_TS_PUSH_EN	R/W	0h	Hardware push 8 enable
14	HW7_TS_PUSH_EN	R/W	0h	Hardware push 7 enable
13	HW6_TS_PUSH_EN	R/W	0h	Hardware push 6 enable
12	HW5_TS_PUSH_EN	R/W	0h	Hardware push 5 enable
11	HW4_TS_PUSH_EN	R/W	0h	Hardware push 4 enable
10	HW3_TS_PUSH_EN	R/W	0h	Hardware push 3 enable
9	HW2_TS_PUSH_EN	R/W	0h	Hardware push 2 enable
8	HW1_TS_PUSH_EN	R/W	0h	Hardware push 1 enable
7	TS_PPM_DIR	R/W	0h	Timestamp PPM Direction
6	TS_COMP_TOG	R/W	0h	Timestamp Compare Toggle mode: 0=TS_COMP is in non-toggle mode, 1=TS_COMP is in toggle mode
5	MODE	R/W	0h	Timestamp mode
4	SEQUENCE_EN	R/W	0h	Sequence Enable
3	TSTAMP_EN	R/W	0h	Host Receive Timestamp Enable
2	TS_COMP_POLARITY	R/W	1h	TS_COMP polarity
1	INT_TEST	R/W	0h	Interrupt test
0	CPTS_EN	R/W	0h	Time sync enable

### 9.6.576 CPTS\_RFTCLK\_SEL\_REG Register (Offset = 0003D008h) [Reset = 00000000h]

CPTS\_RFTCLK\_SEL\_REG is shown in [Table 9-608](#).

Return to the [Summary Table](#).

RFTCLK Select Register

**Table 9-608. CPTS\_RFTCLK\_SEL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	RESERVED	R	0h	
4-0	RFTCLK_SEL	R/W	0h	Reference clock select

### 9.6.577 CPTS\_TS\_PUSH\_REG Register (Offset = 0003D00Ch) [Reset = 00000000h]

CPTS\_TS\_PUSH\_REG is shown in [Table 9-609](#).

Return to the [Summary Table](#).

Time Stamp Event Push Register

**Table 9-609. CPTS\_TS\_PUSH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PUSH	W	0h	Time stamp event push

### 9.6.578 TS\_LOAD\_VAL\_REG Register (Offset = 0003D010h) [Reset = 00000000h]

TS\_LOAD\_VAL\_REG is shown in [Table 9-610](#).

Return to the [Summary Table](#).

Time Stamp Load Low Value Register

**Table 9-610. TS\_LOAD\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TS_LOAD_VAL	R/W	0h	Time stamp load low value



### 9.6.579 CPTS\_TS\_LOAD\_EN\_REG Register (Offset = 0003D014h) [Reset = 00000000h]

CPTS\_TS\_LOAD\_EN\_REG is shown in [Table 9-611](#).

Return to the [Summary Table](#).

Time Stamp Load Enable Register

**Table 9-611. CPTS\_TS\_LOAD\_EN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_LOAD_EN	W	0h	Time stamp load enable

### 9.6.580 TS\_COMP\_VAL\_REG Register (Offset = 0003D018h) [Reset = 00000000h]

TS\_COMP\_VAL\_REG is shown in [Table 9-612](#).

Return to the [Summary Table](#).

Time Stamp Comparison Low Value Register

**Table 9-612. TS\_COMP\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TS_COMP_VAL	R/W	0h	Time stamp comparison low value

### 9.6.581 CPTS\_TS\_COMP\_LEN\_REG Register (Offset = 0003D01Ch) [Reset = 00000000h]

CPTS\_TS\_COMP\_LEN\_REG is shown in [Table 9-613](#).

Return to the [Summary Table](#).

Time Stamp Comparison Length Register

**Table 9-613. CPTS\_TS\_COMP\_LEN\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TS_COMP_LENGTH	R/W	0h	Time stamp comparison length

### 9.6.582 CPTS\_INTSTAT\_RAW\_REG Register (Offset = 0003D020h) [Reset = 00000000h]

CPTS\_INTSTAT\_RAW\_REG is shown in [Table 9-614](#).

Return to the [Summary Table](#).

Interrupt Status Register Raw

**Table 9-614. CPTS\_INTSTAT\_RAW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PEND_RAW	R/W	0h	TS_PEND_RAW int read (before enable)

### 9.6.583 CPTS\_INTSTAT\_MASKED\_REG Register (Offset = 0003D024h) [Reset = 00000000h]

CPTS\_INTSTAT\_MASKED\_REG is shown in [Table 9-615](#).

Return to the [Summary Table](#).

Interrupt Status Register Masked

**Table 9-615. CPTS\_INTSTAT\_MASKED\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PEND	R	0h	TS_PEND masked interrupt read (after enable)

### 9.6.584 CPTS\_INT\_ENABLE\_REG Register (Offset = 0003D028h) [Reset = 00000000h]

CPTS\_INT\_ENABLE\_REG is shown in [Table 9-616](#).

Return to the [Summary Table](#).

Interrupt Enable Register

**Table 9-616. CPTS\_INT\_ENABLE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	TS_PEND_EN	R/W	0h	TS_PEND masked interrupt enable

### 9.6.585 CPTS\_TS\_COMP\_NUDGE\_REG Register (Offset = 0003D02Ch) [Reset = 00000000h]

CPTS\_TS\_COMP\_NUDGE\_REG is shown in [Table 9-617](#).

Return to the [Summary Table](#).

Time Stamp Comparison Nudge Register

**Table 9-617. CPTS\_TS\_COMP\_NUDGE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	NUDGE	R/W	0h	This 2s complement number is added to the ts_comp_length value to increase or decrease the TS_COMP length by the nudge amount

### 9.6.586 CPTS\_EVENT\_POP\_REG Register (Offset = 0003D030h) [Reset = 00000000h]

CPTS\_EVENT\_POP\_REG is shown in [Table 9-618](#).

Return to the [Summary Table](#).

Event Pop Register

**Table 9-618. CPTS\_EVENT\_POP\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	EVENT_POP	W	0h	Event pop



### 9.6.587 CPTS\_EVENT\_0\_REG Register (Offset = 0003D034h) [Reset = 00000000h]

CPTS\_EVENT\_0\_REG is shown in [Table 9-619](#).

Return to the [Summary Table](#).

Event 0 Register

**Table 9-619. CPTS\_EVENT\_0\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIME_STAMP	R	0h	Time Stamp

### 9.6.588 CPTS\_EVENT\_1\_REG Register (Offset = 0003D038h) [Reset = 00000000h]

CPTS\_EVENT\_1\_REG is shown in [Table 9-620](#).

Return to the [Summary Table](#).

Event 1 Register

**Table 9-620. CPTS\_EVENT\_1\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R	0h	
29	PREMPT_QUEUE	R	0h	Prempt QUEUE
28-24	PORT_NUMBER	R	0h	Port number
23-20	EVENT_TYPE	R	0h	Event type
19-16	MESSAGE_TYPE	R	0h	Message type
15-0	SEQUENCE_ID	R	0h	Sequence ID

### 9.6.589 CPTS\_EVENT\_2\_REG Register (Offset = 0003D03Ch) [Reset = 00000000h]

CPTS\_EVENT\_2\_REG is shown in [Table 9-621](#).

Return to the [Summary Table](#).

Event 2 Register

**Table 9-621. CPTS\_EVENT\_2\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	DOMAIN	R	0h	Domain

### 9.6.590 CPTS\_EVENT\_3\_REG Register (Offset = 0003D040h) [Reset = 00000000h]

CPTS\_EVENT\_3\_REG is shown in [Table 9-622](#).

Return to the [Summary Table](#).

Event 3 Register

**Table 9-622. CPTS\_EVENT\_3\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TIME_STAMP	R	0h	Time Stamp

### 9.6.591 CPTS\_TS\_LOAD\_HIGH\_VAL\_REG Register (Offset = 0003D044h) [Reset = 00000000h]

CPTS\_TS\_LOAD\_HIGH\_VAL\_REG is shown in [Table 9-623](#).

Return to the [Summary Table](#).

Time Stamp Load High Value Register

**Table 9-623. CPTS\_TS\_LOAD\_HIGH\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TS_LOAD_VAL	R/W	0h	Time stamp load high value

### 9.6.592 CPTS\_TS\_COMP\_HIGH\_VAL\_REG Register (Offset = 0003D048h) [Reset = 00000000h]

CPTS\_TS\_COMP\_HIGH\_VAL\_REG is shown in [Table 9-624](#).

Return to the [Summary Table](#).

Time Stamp Comparison High Value Register

**Table 9-624. CPTS\_TS\_COMP\_HIGH\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TS_COMP_HIGH_VAL	R/W	0h	Time stamp comparison high value

### 9.6.593 CPTS\_TS\_ADD\_VAL\_REG Register (Offset = 0003D04Ch) [Reset = 00000000h]

CPTS\_TS\_ADD\_VAL\_REG is shown in [Table 9-625](#).

Return to the [Summary Table](#).

TS Add Value Register

**Table 9-625. CPTS\_TS\_ADD\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	
2-0	ADD_VAL	R/W	0h	Add Value

### 9.6.594 CPTS\_TS\_PPM\_LOW\_VAL\_REG Register (Offset = 0003D050h) [Reset = 00000000h]

CPTS\_TS\_PPM\_LOW\_VAL\_REG is shown in [Table 9-626](#).

Return to the [Summary Table](#).

Time Stamp PPM Low Value Register

**Table 9-626. CPTS\_TS\_PPM\_LOW\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TS_PPM_LOW_VAL	R/W	0h	Time stamp PPM Low value



### 9.6.595 CPTS\_TS\_PPM\_HIGH\_VAL\_REG Register (Offset = 0003D054h) [Reset = 00000000h]

CPTS\_TS\_PPM\_HIGH\_VAL\_REG is shown in [Table 9-627](#).

Return to the [Summary Table](#).

Time Stamp PPM High Value Register

**Table 9-627. CPTS\_TS\_PPM\_HIGH\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	TS_PPM_HIGH_VAL	R/W	0h	Time stamp PPM High value

### 9.6.596 CPTS\_TS\_NUDGE\_VAL\_REG Register (Offset = 0003D058h) [Reset = 00000000h]

CPTS\_TS\_NUDGE\_VAL\_REG is shown in [Table 9-628](#).

Return to the [Summary Table](#).

Time Stamp Nudge Value Register

**Table 9-628. CPTS\_TS\_NUDGE\_VAL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	TS_NUDGE_VAL	R/W	0h	Time stamp Nudge value

### 9.6.597 CPTS\_TS\_CONFIG Register (Offset = 0003D0D0h) [Reset = 00002003h]

CPTS\_TS\_CONFIG is shown in [Table 9-629](#).

Return to the [Summary Table](#).

Time Stamp Configuration Read

**Table 9-629. CPTS\_TS\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-8	EVNT_FIFO_DEPTH	R	20h	The Event FIFO Depth
7-0	NUM_GENF	R	3h	The number of CPTS GENF outputs

### 9.6.598 TS\_GENF0\_COMP\_LOW\_REG Register (Offset = 0003D0E0h) [Reset = 00000000h]

TS\_GENF0\_COMP\_LOW\_REG is shown in [Table 9-630](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Comparison Low Value

**Table 9-630. TS\_GENF0\_COMP\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_LOW	R/W	0h	Time Stamp Generate Function Comparison Low Value

### 9.6.599 TS\_GENF0\_COMP\_HIGH\_REG Register (Offset = 0003D0E4h) [Reset = 00000000h]

TS\_GENF0\_COMP\_HIGH\_REG is shown in [Table 9-631](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Comparison high Value

**Table 9-631. TS\_GENF0\_COMP\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_HIGH	R/W	0h	Time Stamp Generate Function Comparison High Value

### 9.6.600 TS\_GENF0\_CONTROL\_REG Register (Offset = 0003D0E8h) [Reset = 00000000h]

TS\_GENF0\_CONTROL\_REG is shown in [Table 9-632](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Control

**Table 9-632. TS\_GENF0\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	POLARITY_INV	R/W	0h	Time Stamp Generate Function Polarity Invert
0	PPM_DIR	R/W	0h	Time Stamp Generate Function PPM Direction

### 9.6.601 TS\_GENF0\_LENGTH\_REG Register (Offset = 0003D0ECh) [Reset = 00000000h]

TS\_GENF0\_LENGTH\_REG is shown in [Table 9-633](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Length Value

**Table 9-633. TS\_GENF0\_LENGTH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LENGTH	R/W	0h	Time Stamp Generate Function Length Value

### 9.6.602 TS\_GENF0\_PPM\_LOW\_REG Register (Offset = 0003D0F0h) [Reset = 00000000h]

TS\_GENF0\_PPM\_LOW\_REG is shown in [Table 9-634](#).

Return to the [Summary Table](#).

Time Stamp Generate Function PPM Low Value

**Table 9-634. TS\_GENF0\_PPM\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PPM_LOW	R/W	0h	Time Stamp Generate Function PPM Low Value



### 9.6.603 TS\_GENF0\_PPM\_HIGH\_REG Register (Offset = 0003D0F4h) [Reset = 00000000h]

TS\_GENF0\_PPM\_HIGH\_REG is shown in [Table 9-635](#).

Return to the [Summary Table](#).

Time Stamp Generate Function PPM High Value

**Table 9-635. TS\_GENF0\_PPM\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	PPM_HIGH	R/W	0h	Time Stamp Generate Function PPM High Value

### 9.6.604 TS\_GENF0\_NUDGE\_REG Register (Offset = 0003D0F8h) [Reset = 00000000h]

TS\_GENF0\_NUDGE\_REG is shown in [Table 9-636](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Nudge Value

**Table 9-636. TS\_GENF0\_NUDGE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	NUDGE	R/W	0h	Time Stamp Generate Function Nudge Value

### 9.6.605 TS\_GENF1\_COMP\_LOW\_REG Register (Offset = 0003D100h) [Reset = 00000000h]

TS\_GENF1\_COMP\_LOW\_REG is shown in [Table 9-637](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Comparison Low Value

**Table 9-637. TS\_GENF1\_COMP\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_LOW	R/W	0h	Time Stamp Generate Function Comparison Low Value

### 9.6.606 TS\_GENF1\_COMP\_HIGH\_REG Register (Offset = 0003D104h) [Reset = 00000000h]

TS\_GENF1\_COMP\_HIGH\_REG is shown in [Table 9-638](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Comparison high Value

**Table 9-638. TS\_GENF1\_COMP\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_HIGH	R/W	0h	Time Stamp Generate Function Comparison High Value

### 9.6.607 TS\_GENF1\_CONTROL\_REG Register (Offset = 0003D108h) [Reset = 00000000h]

TS\_GENF1\_CONTROL\_REG is shown in [Table 9-639](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Control

**Table 9-639. TS\_GENF1\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	POLARITY_INV	R/W	0h	Time Stamp Generate Function Polarity Invert
0	PPM_DIR	R/W	0h	Time Stamp Generate Function PPM Direction

### 9.6.608 TS\_GENF1\_LENGTH\_REG Register (Offset = 0003D10Ch) [Reset = 00000000h]

TS\_GENF1\_LENGTH\_REG is shown in [Table 9-640](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Length Value

**Table 9-640. TS\_GENF1\_LENGTH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LENGTH	R/W	0h	Time Stamp Generate Function Length Value

### 9.6.609 TS\_GENF1\_PPM\_LOW\_REG Register (Offset = 0003D110h) [Reset = 00000000h]

TS\_GENF1\_PPM\_LOW\_REG is shown in [Table 9-641](#).

Return to the [Summary Table](#).

Time Stamp Generate Function PPM Low Value

**Table 9-641. TS\_GENF1\_PPM\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PPM_LOW	R/W	0h	Time Stamp Generate Function PPM Low Value

### 9.6.610 TS\_GENF1\_PPM\_HIGH\_REG Register (Offset = 0003D114h) [Reset = 00000000h]

TS\_GENF1\_PPM\_HIGH\_REG is shown in [Table 9-642](#).

Return to the [Summary Table](#).

Time Stamp Generate Function PPM High Value

**Table 9-642. TS\_GENF1\_PPM\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	PPM_HIGH	R/W	0h	Time Stamp Generate Function PPM High Value



### 9.6.611 TS\_GENF1\_NUDGE\_REG Register (Offset = 0003D118h) [Reset = 00000000h]

TS\_GENF1\_NUDGE\_REG is shown in [Table 9-643](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Nudge Value

**Table 9-643. TS\_GENF1\_NUDGE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	NUDGE	R/W	0h	Time Stamp Generate Function Nudge Value

### 9.6.612 TS\_GENF2\_COMP\_LOW\_REG Register (Offset = 0003D120h) [Reset = 00000000h]

TS\_GENF2\_COMP\_LOW\_REG is shown in [Table 9-644](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Comparison Low Value

**Table 9-644. TS\_GENF2\_COMP\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_LOW	R/W	0h	Time Stamp Generate Function Comparison Low Value

### 9.6.613 TS\_GENF2\_COMP\_HIGH\_REG Register (Offset = 0003D124h) [Reset = 00000000h]

TS\_GENF2\_COMP\_HIGH\_REG is shown in [Table 9-645](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Comparison high Value

**Table 9-645. TS\_GENF2\_COMP\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_HIGH	R/W	0h	Time Stamp Generate Function Comparison High Value

### 9.6.614 TS\_GENF2\_CONTROL\_REG Register (Offset = 0003D128h) [Reset = 00000000h]

TS\_GENF2\_CONTROL\_REG is shown in [Table 9-646](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Control

**Table 9-646. TS\_GENF2\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	POLARITY_INV	R/W	0h	Time Stamp Generate Function Polarity Invert
0	PPM_DIR	R/W	0h	Time Stamp Generate Function PPM Direction

### 9.6.615 TS\_GENF2\_LENGTH\_REG Register (Offset = 0003D12Ch) [Reset = 00000000h]

TS\_GENF2\_LENGTH\_REG is shown in [Table 9-647](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Length Value

**Table 9-647. TS\_GENF2\_LENGTH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LENGTH	R/W	0h	Time Stamp Generate Function Length Value

### 9.6.616 TS\_GENF2\_PPM\_LOW\_REG Register (Offset = 0003D130h) [Reset = 00000000h]

TS\_GENF2\_PPM\_LOW\_REG is shown in [Table 9-648](#).

Return to the [Summary Table](#).

Time Stamp Generate Function PPM Low Value

**Table 9-648. TS\_GENF2\_PPM\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PPM_LOW	R/W	0h	Time Stamp Generate Function PPM Low Value

### 9.6.617 TS\_GENF2\_PPM\_HIGH\_REG Register (Offset = 0003D134h) [Reset = 00000000h]

TS\_GENF2\_PPM\_HIGH\_REG is shown in [Table 9-649](#).

Return to the [Summary Table](#).

Time Stamp Generate Function PPM High Value

**Table 9-649. TS\_GENF2\_PPM\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	PPM_HIGH	R/W	0h	Time Stamp Generate Function PPM High Value

### 9.6.618 TS\_GENF2\_NUDGE\_REG Register (Offset = 0003D138h) [Reset = 00000000h]

TS\_GENF2\_NUDGE\_REG is shown in [Table 9-650](#).

Return to the [Summary Table](#).

Time Stamp Generate Function Nudge Value

**Table 9-650. TS\_GENF2\_NUDGE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	NUDGE	R/W	0h	Time Stamp Generate Function Nudge Value



### 9.6.619 TS\_ESTF\_COMP\_LOW\_REG Register (Offset = 0003D200h) [Reset = 00000000h]

TS\_ESTF\_COMP\_LOW\_REG is shown in [Table 9-651](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function Comparison Low Value

**Table 9-651. TS\_ESTF\_COMP\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_LOW	R/W	0h	Time Stamp ESTF Generate Function Comparison Low Value

### 9.6.620 TS\_ESTF\_COMP\_HIGH\_REG Register (Offset = 0003D204h) [Reset = 00000000h]

TS\_ESTF\_COMP\_HIGH\_REG is shown in [Table 9-652](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function Comparison high Value

**Table 9-652. TS\_ESTF\_COMP\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP_HIGH	R/W	0h	Time Stamp ESTF Generate Function Comparison High Value

### 9.6.621 TS\_ESTF\_CONTROL\_REG Register (Offset = 0003D208h) [Reset = 00000000h]

TS\_ESTF\_CONTROL\_REG is shown in [Table 9-653](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function Control

**Table 9-653. TS\_ESTF\_CONTROL\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	POLARITY_INV	R/W	0h	Time Stamp ESTF Generate Function Polarity Invert
0	PPM_DIR	R/W	0h	Time Stamp ESTF Generate Function PPM Direction

### 9.6.622 TS\_ESTF\_LENGTH\_REG Register (Offset = 0003D20Ch) [Reset = 00000000h]

TS\_ESTF\_LENGTH\_REG is shown in [Table 9-654](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function Length Value

**Table 9-654. TS\_ESTF\_LENGTH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LENGTH	R/W	0h	Time Stamp ESTF Generate Function Length Value

### 9.6.623 TS\_ESTF\_PPM\_LOW\_REG Register (Offset = 0003D210h) [Reset = 00000000h]

TS\_ESTF\_PPM\_LOW\_REG is shown in [Table 9-655](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function PPM Low Value

**Table 9-655. TS\_ESTF\_PPM\_LOW\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PPM_LOW	R/W	0h	Time Stamp ESTF Generate Function PPM Low Value

### 9.6.624 TS\_ESTF\_PPM\_HIGH\_REG Register (Offset = 0003D214h) [Reset = 00000000h]

TS\_ESTF\_PPM\_HIGH\_REG is shown in [Table 9-656](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function PPM High Value

**Table 9-656. TS\_ESTF\_PPM\_HIGH\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	0h	
9-0	PPM_HIGH	R/W	0h	Time Stamp ESTF Generate Function PPM High Value

### 9.6.625 TS\_ESTF\_NUDGE\_REG Register (Offset = 0003D218h) [Reset = 00000000h]

TS\_ESTF\_NUDGE\_REG is shown in [Table 9-657](#).

Return to the [Summary Table](#).

Time Stamp ESTF Generate Function Nudge Value

**Table 9-657. TS\_ESTF\_NUDGE\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	NUDGE	R/W	0h	Time Stamp ESTF Generate Function Nudge Value

### 9.6.626 ALE\_MOD\_VER Register (Offset = 0003E000h) [Reset = 00290105h]

ALE\_MOD\_VER is shown in [Table 9-658](#).

Return to the [Summary Table](#).

The Module and Version Register identifies the module identifier and revision of the ALE\_2g32 module.

**Table 9-658. ALE\_MOD\_VER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	MODULE_ID	R	29h	ALE_2g32 module ID.
15-11	RTL_VERSION	R	0h	RTL Version.
10-8	MAJOR_REVISION	R	1h	Major Revision.
7-6	CUSTOM_REVISION	R	0h	Custom Revision.
5-0	MINOR_REVISION	R	5h	Minor Revision.



### 9.6.627 ALE\_ALE\_STATUS Register (Offset = 0003E004h) [Reset = XXXX00X0h]

ALE\_ALE\_STATUS is shown in [Table 9-659](#).

Return to the [Summary Table](#).

The ALE status provides information on the ALE configuration and state. The ~iramdepth is used to determine how IPv6 entries are stored in the table. IPv6 entries are stored in two entries where IPv6 Entry hi is designated by the odd slice index and lo is designated by the even slice index. The slice index is above the ram depth like {SlixelIndex,RamIndex}. So for a 64 deep RAM index of 0x005, the Hi portion of the IPv6 entry is located at 0x005|0x040 and the Lo portion is located at 0x005&(~0x040).

**Table 9-659. ALE\_ALE\_STATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	UREGANDREGMSK12	R	0h	When set, the unregistered multicast field is a mask versus an index on 12 bit boundary in the ALE table.
30	UREGANDREGMSK08	R	0h	When set, the unregistered multicast field is a mask versus an index on 8 bit boundary in the ALE table.
29-16	RESERVED	R	0h	
15-8	POLCNTDIV8	R	0h	This is the number of Classifiers the ALE implements divided by 8. A value of 4 indicates 32 policer engines total.
7	RAMDEPTH128	R	0h	The number of ALE entries per slice of the table when this is set it indicates the depth is 128 if both ramdepth128 and ramdepth32 are zero the depth is 64.
6	RAMDEPTH32	R	0h	The number of ALE entries per slice of the table when this is set it indicates the depth is 32 if both ramdepth128 and ramdepth32 are zero the depth is 64.
5	RESERVED	R	0h	
4-0	KLUENTRIES	R	0h	This is the number of table entries total divided by 1024. A value of 1 indicates 1024 table entries. A value of 8 indicates 8192 table entries.

### 9.6.628 ALE\_ALE\_CONTROL Register (Offset = 0003E008h) [Reset = XXXXXX00h]

ALE\_ALE\_CONTROL is shown in [Table 9-660](#).

Return to the [Summary Table](#).

The ALE Control Register is used to set the ALE modes used for all ports.

**Table 9-660. ALE\_ALE\_CONTROL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	ENABLE_ALE	R/W	0h	Enable ALE 0 - Drop all packets 1 - Enable ALE packet processing
30	CLEAR_TABLE	R/W	0h	Clear ALE address table - Setting this bit causes the ALE hardware to write all table bit values to zero. Software must perform a clear table operation as part of the ALE setup/configuration process. Setting this bit causes all ALE accesses to be held up for 64 clocks while the clear is performed. Access to all ALE registers will be blocked (wait states) until the 64 clocks have completed. This bit cannot be read as one because the read is blocked until the clear table is completed at which time this bit is cleared to zero.
29	AGE_OUT_NOW	R/W	0h	Age Out Address Table Now - Setting this bit causes the ALE hardware to remove (free up) any ageable table entry that does not have a set touch bit. This bit is cleared when the age out process has completed. This bit may be read. The age out process takes four times the number of table entries clock cycles (4096 cycles for 1K addresses) best case (no ale packet processing during ageout) and sixty five times the number of table entries clock cycles (66560 cycles for 1K addresses) absolute worst case.
28-25	RESERVED	R	0h	
24	MIRROR_DP	R/W	0h	Mirror Destination Port - This field defines the port to which destination traffic destined will be duplicated. That is all traffic that is forwarded to this port will also be mirrored to the ~imirror_top port.
23-21	UPD_BW_CTRL	R/W	0h	The ~iupd_bw_ctrl field allows for up to 8 times the rate in which adds, updates, touches, writes, and aging updates can occur. At frequencies of 350Mhz, the table update rate should be at it lowest or 5 Million updates per second. When operating the switch core at frequencies or above, the ~iupd_bw_ctrl can be programmed more aggressive. If the ~iupd_bw_ctrl is set but the frequency of the switch subsystem is below the associated value, ALE will drop packets due to insufficient time to complete lookup under high traffic loads. 0 - 350Mhz, 5M 1 - 359Mhz, 11M 2 - 367Mhz, 16M 3 - 375Mhz, 22M 4 - 384Mhz, 28M 5 - 392Mhz, 34M 6 - 400Mhz, 39M 7 - 409Mhz, 45M
20-17	RESERVED	R	0h	
16	MIRROR_TOP	R/W	0h	Mirror To Port - This field defines the destination port for the mirror traffic. If the traffic is received or transmitted on the mirror destination port it will not be duplicated. Traffic defined as mirror traffic only may be dropped by the switch due to congestion.

**Table 9-660. ALE\_ALE\_CONTROL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
15	UPD_STATIC	R/W	0h	Update Static Entries - A static Entry is an entry that is not agable. When clear this bit will prevent any static entry (agable bit clear) from being updated due to port change. When set it allows static entries (agable bit clear) to update the source port if required. This bit should normally be '0' for most switch configurations.
14	RESERVED	R	0h	
13	UVLAN_NO_LEARN	R/W	0h	Unknown VLAN No Learn - This field when set will prevent source addresses of unknown VLAN IDs from being automatically added into the look up table if learning is enabled.
12	MIRROR_MEN	R/W	0h	Mirror Match Entry Enable - This field enables the match mirror option. When this bit is set any traffic whose destination, source, VLAN or OUI matches the ~imirror_midx entry index will have that traffic also sent to the ~imirror_top port.
11	MIRROR_DEN	R/W	0h	Mirror Destination Port Enable - This field enables the destination port mirror option. When this bit is set any traffic destined for the ~imirror_dp port will have its transmit traffic also sent to the ~imirror_top port.
10	MIRROR_SEN	R/W	0h	Mirror Source Port Enable - This field enables the source port mirror option. When this bit is set any port with the ~ipX_mirror_sp set in the ALE Port Control registers set will have its received traffic also sent to the ~imirror_top port.
9	RESERVED	R	0h	
8	EN_HOST_UNI_FLOOD	R/W	0h	Unknown unicast packets flood to host 0 - unknown unicast packets are not sent to the host 1 - unknown unicast packets flood to host port as well as other ports
7	LEARN_NO_VLANID	R/W	0h	Learn No VID - 0 - VID is learned with the source address 1 - VID is not learned with the source address (source address is not tied to VID). Determines the entry type.
6	ENABLE_VID0_MODE	R/W	0h	Enable VLAN ID = 0 Mode 0 - Process the priority tagged packet with VID = PORT_VLAN[11:0]. 1 - Process the priority tagged packet with VID = 0.
5	ENABLE_OUI_DENY	R/W	0h	Enable OUI Deny Mode - When set, any packet with a non-matching OUI source address will be dropped to the host unless the packet destination address matches a supervisory destination address table entry. When cleared, any packet source address matching an OUI address table entry will be dropped to the host unless the destination address matches with a supervisory destination address table entry.
4	ENABLE_BYPASS	R/W	0h	ALE Bypass - When set, packets received on non-host ports are sent to the host. It is expected that packets from the host are directed to the particular port. 0 - no bypass 1 - bypass the ALE
3	BCAST_MCAST_CTL	R/W	0h	Rate Limit Transmit mode 0 - Broadcast and multicast rate limit counters are received port based 1 - Broadcast and multicast rate limit counters are transmit port based
2	ALE_VLAN_AWARE	R/W	0h	ALE VLAN Aware - Determines how traffic is forwarded using VLAN rules. 0 - Simple switch rules, packets forwarded to all ports for unknown destinations. 1 - VLAN Aware rules, packets forwarded based on VLAN members

**Table 9-660. ALE\_ALE\_CONTROL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	ENABLE_AUTH_MODE	R/W	0h	Enable MAC Authorization Mode - Mac authorization mode requires that all table entries be made by the host software. There is no auto learning of addresses in authorization mode and the packet will be dropped if the source address is not found (and the destination address is not a multicast address with the super table entry bit set). 0 - The ALE is not in MAC authorization mode 1 - The ALE is in MAC authorization mode
0	ENABLE_RATE_LIMIT	R/W	0h	Enable Broadcast and Multicast Rate Limit 0 - Broadcast/Multicast rates not limited 1 - Broadcast/Multicast packet reception limited to the port control register rate limit fields.

### 9.6.629 ALE\_ALE\_CTRL2 Register (Offset = 0003E00Ch) [Reset = XX0XXX0h]

ALE\_ALE\_CTRL2 is shown in [Table 9-661](#).

Return to the [Summary Table](#).

The ALE Control 2 Register is used to set the extended features used for all ports.

**Table 9-661. ALE\_ALE\_CTRL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	TRK_EN_DST	R/W	0h	Trunk Enable Destination Address - This field enables the destination MAC address to be used with the hash function $G(X) = 1 + X + X^3$ and affect the trunk port transmit link determination.
30	TRK_EN_SRC	R/W	0h	Trunk Enable Source Address - This field enables the source MAC address to be used with the hash function $G(X) = 1 + X + X^3$ and affect the trunk port transmit link determination.
29	TRK_EN_PRI	R/W	0h	Trunk Enable Priority - This field enables the VLAN Priority bits to be used with the hash function $G(X) = 1 + X + X^3$ and affect the trunk port transmit link determination. In the event that DSCP mapping is enabled and there is no VLAN the DSCP priority will be used. For all other non IP frames without VLAN the port default priority is used.
28	RESERVED	R	0h	
27	TRK_EN_IVLAN	R/W	0h	Trunk Enable Inner VLAN - This field enables the inner VLAN ID value (C-VLANID) to be used with the hash function $G(X) = 1 + X + X^3$ and affect the trunk port transmit link determination.
26	RESERVED	R	0h	
25	TRK_EN_SIP	R/W	0h	Trunk Enable Source IP Address - This field enables the source IP address to be used with the hash function $G(X) = 1 + X + X^3$ and affect the trunk port transmit link determination. This feature supports No tag, Priority tagged, VLAN tagged, Q-in-Q double tagging for both IPV6 and IPV4.
24	TRK_EN_DIP	R/W	0h	Trunk Enable Destination IP Address - This field enables the destination IP address to be used with the hash function $G(X) = 1 + X + X^3$ and affect the trunk port transmit link determination. This feature supports No tag, Priority tagged, VLAN tagged, Q-in-Q double tagging for both IPV6 and IPV4.
23	DROP_BADLEN	R/W	0h	Drop Bad Length will drop any packet that the 802.3 length field is larger than the packet. Ethertypes 0-1500 are 802.3 lengths, all others are Ether types.
22	NODROP_SRCMCST	R/W	0h	No Drop Source Multicast will disable the dropping of any source address with the multicast bit set.
21	DEFNOFRAG	R/W	0h	Default No Frag field will cause an IPv4 fragmented packet to be dropped if a VLAN entry is not found.
20	DEFLMTNXTHDR	R/W	0h	Default limit next header field will cause an IPv4 protocol or IPv6 next header packet to be dropped if a VLAN entry is not found and the protocol or next header does not match the ~iALE_NXT_HDR register values.
19	RESERVED	R	0h	

**Table 9-661. ALE\_ALE\_CTRL2 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
18-16	TRK_BASE	R/W	0h	<p>Trunk Base - This field is the hash formula starting value. Changing this value will cause the packet distribution on trunk ports to be changed.</p> <p>If all the ~itrk_en_dst, ~itrk_en_src, ~itrk_en_pri and ~itrk_en_vlan are '0', this value is used as the distribution index. That is a '0' will select the 1st bit of an 'N' link trunk, a '1' will select the second, etc.</p> <p>Below is the distribution across the trunk links.</p> <p>The first number in the ~iitalic sequence indicates the traffic is sent to the lowest numbered port of a trunk group.</p> <p>For example if you have a 3 port trunk, the hash result 0 will go to the base port (0), hash result 1 will go to the highest port of the trunk group (2), hash result 2 will go to the middle port (1), etc.</p> <p>1 - ~i00000000  2 - ~i01010101  3 - ~i02102102  4 - ~i03210321</p>
15	MULTIHOST	R/W	0h	The ~multihost allows host traffic to be sent back to the host if the DA is market for the host port.
14-5	RESERVED	R	0h	
4-0	MIRROR_MIDX	R/W	0h	<p>Mirror Index - This field is the ALE lookup table entry index that when a match occurs will cause this traffic to be mirrored to the ~imirror_top port.</p> <p>That is any VLAN, ONU or address with or withou VLAN can be selected for traffic mirroring.</p>

### 9.6.630 ALE\_ALE\_PRESCALE Register (Offset = 0003E010h) [Reset = 00000000h]

ALE\_ALE\_PRESCALE is shown in [Table 9-662](#).

Return to the [Summary Table](#).

The ALE Prescale Register is used to set the Broadcast and Multicast rate limiting prescaler value.

**Table 9-662. ALE\_ALE\_PRESCALE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	ALE_PRESCALE	R/W	0h	ALE Prescale - The input clock is divided by this value for use in the multicast/broadcast rate limiters. The minimum operating value is 0x10. The prescaler is off when the value is zero.

### 9.6.631 ALE\_ALE\_AGING\_CTRL Register (Offset = 0003E014h) [Reset = XX000000h]

ALE\_ALE\_AGING\_CTRL is shown in [Table 9-663](#).

Return to the [Summary Table](#).

The ALE Aging Control sets the aging interval which will cause periodic aging to occur. This value specifies the minimum time between aging starts.

**Table 9-663. ALE\_ALE\_AGING\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PRESCALE_2_DISABLE	R/W	0h	ALE Prescaler 2 Disable - When set will divide the aging interval by 1000. This bit is designed for device verification and should not be used in production software. Combination of PreScale1Disable and PreScale2Disable will divide the aging interval by 1,000,000 for test purposes.
30	PRESCALE_1_DISABLE	R/W	0h	ALE Prescaler 1 Disable - When set will divide the aging interval by 1000. This bit is designed for device verification and should not be used in production software. Combination of PreScale1Disable and PreScale2Disable will divide the aging interval by 1,000,000 for test purposes.
29-24	RESERVED	R	0h	
23-0	ALE_AGING_TIMER	R/W	0h	ALE Aging Timer - This field specifies the number of clock cycles times 1,000,000 between aging operations.



### 9.6.632 ALE\_ALE\_NXT\_HDR Register (Offset = 0003E01Ch) [Reset = 00000000h]

ALE\_ALE\_NXT\_HDR is shown in [Table 9-664](#).

Return to the [Summary Table](#).

The ALE Next Header is used to limit the IPv6 Next header or IPv4 Protocol values found in the IP header. It is enabled via the ~iLmtNxtHdr bit in the VLAN entry. All four ~iip\_nxt\_hdr0-3 are compared when enabled, so if only one is required, set them all to the one value to be tested.

**Table 9-664. ALE\_ALE\_NXT\_HDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	IP_NXT_HDR3	R/W	0h	The ~iip_nxt_hdr3 is the forth protocol or next header compared when enabled.
23-16	IP_NXT_HDR2	R/W	0h	The ~iip_nxt_hdr2 is the third protocol or next header compared when enabled.
15-8	IP_NXT_HDR1	R/W	0h	The ~iip_nxt_hdr1 is the second protocol or next header compared when enabled.
7-0	IP_NXT_HDR0	R/W	0h	The ~iip_nxt_hdr0 is the first protocol or next header compared when enabled.

### 9.6.633 ALE\_ALE\_TBLCTL Register (Offset = 0003E020h) [Reset = XXXXXX0h]

ALE\_ALE\_TBLCTL is shown in [Table 9-665](#).

Return to the [Summary Table](#).

The ALE table control register is used to read or write that ALE table entries. After writing to this register any read or write to any ALE register will be stalled until the read or write operation completes.

**Table 9-665. ALE\_ALE\_TBLCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	TABLEWR	R/W	0h	Table Write - This bit is used to write the table words to the lookup table. 0 - Table Read Operation is performed. The contents of the ~b TABLEIDX entry will be read into the ~b ALE_TBLWx registers 1 - Table write operation is performed. This will take the current contents from the ~b ALE_TBLWx registers and write them to the table at the specified ~b TABLEIDX.
30-5	RESERVED	R	0h	
4-0	TABLEIDX	R/W	0h	The table index is used to determine which lookup table entry is read or written.

### 9.6.634 ALE\_ALE\_TBLW2 Register (Offset = 0003E034h) [Reset = 00000000h]

ALE\_ALE\_TBLW2 is shown in [Table 9-666](#).

Return to the [Summary Table](#).

The ALE Table Word 2 is the most significant word of an ALE table entry.

**Table 9-666. ALE\_ALE\_TBLW2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R	0h	
6-0	TABLEWRD2	R/W	0h	Table Entry bits [71:64]

### 9.6.635 ALE\_ALE\_TBLW1 Register (Offset = 0003E038h) [Reset = 00000000h]

ALE\_ALE\_TBLW1 is shown in [Table 9-667](#).

Return to the [Summary Table](#).

The ALE Table Word 1 is the middle word of an ALE table entry.

**Table 9-667. ALE\_ALE\_TBLW1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TABLEWRD1	R/W	0h	Table Entry bits [63:32]

### 9.6.636 ALE\_ALE\_TBLW0 Register (Offset = 0003E03Ch) [Reset = 00000000h]

ALE\_ALE\_TBLW0 is shown in [Table 9-668](#).

Return to the [Summary Table](#).

The ALE Table Word 0 is the least significant word of an ALE table entry.

**Table 9-668. ALE\_ALE\_TBLW0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TABLEWRD0	R/W	0h	Table Entry bits [31:0]

### 9.6.637 ALE\_I0\_ALE\_PORTCTL0\_0 Register (Offset = 0003E040h) [Reset = 000000X0h]

ALE\_I0\_ALE\_PORTCTL0\_0 is shown in [Table 9-669](#).

Return to the [Summary Table](#).

The ALE Port Control Register sets the port specific modes of operation.

**Table 9-669. ALE\_I0\_ALE\_PORTCTL0\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	I0_REG_P0_BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field.
23-16	I0_REG_P0_MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field. The ~imcast_limit is the number of Multicast packets that will be forwarded per ~iale_prescale time.
15	I0_REG_P0_DROP_DOUBLE_VLAN	R/W	0h	Drop Double VLAN - When set cause any received packet with double VLANs to be dropped. That is if there are two ctag or two stag fields in the packet it will be dropped.
14	I0_REG_P0_DROP_DUAL_VLAN	R/W	0h	Drop Dual VLAN - When set will cause any received packet with dual VLAN stag followed by ctag to be dropped.
13	I0_REG_P0_MACONLY_CAF	R/W	0h	Mac Only Copy All Frames - When set a Mac Only port will transfer all received good frames to the host. When clear a Mac Only port will transfer packets to the host based on ALE destination address lookup operation (which operates more like an Ethernet Mac). A Mac Only port is a port with ~imaconly set.
12	I0_REG_P0_DISABLE_PAUTH_MOD	R/W	0h	Disable Port authorization - When set will allow unknown addresses to arrive on a switch in authorization mode. It is intended for device to device network connection on ports which do not require MACSEC encryption.
11	I0_REG_P0_MACONLY	R/W	0h	MAC Only - When set enables this port be treated like a MAC port for the host. All traffic received is only sent to the host. The host must direct traffic to this port as the lookup engine will not send traffic to the ports with the ~ip0_maconly bit set and the ~ip0_no_learn also set. If ~ip0_maconly bit is set and the ~ip0_no_learn is not set, the host can send non-directed packets that can be sent to the destination of a MacOnly port. It is also possible that The host can broadcast to all ports including MacOnly ports in this mode.
10	I0_REG_P0_TRUNKEN	R/W	0h	Trunk Enable - This field is used to enable a port into a trunk. Any port can be used as a trunk port, any two or more ports with the ~ip0_trunken its set and having the same ~ip0_trunknum will be placed in the same trunk. There is no requirement for trunk ports to be adjacent. If all ports are enabled in the same trunk, no traffic can flow as traffic received within a trunk is never transmitted out the same trunk. If only a single port is a member of a trunk, it looks like a normal port with exception of entries in the look up table will be noted as a trunk entry.

**Table 9-669. ALE\_I0\_ALE\_PORTCTL0\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9-8	I0_REG_P0_TRUNKNUM	R/W	0h	Trunk Number - This field is used as the trunk number when the ~ip0_trunken is also set. Ports with the same trunk number that have the ~ip0_trunken also set will have traffic distributed within the trunk based on the result of the hash function described above.
7	I0_REG_P0_MIRROR_SP	R/W	0h	Mirror Source Port - This field enables the source port mirror option. When this bit is set any traffic received on the port with the reg_p0_mirror_sp bit set will have its received traffic also sent to the ~imirror_top port.
6	RESERVED	R	0h	
5	I0_REG_P0_NO_SA_UPDATE	R/W	0h	No Source Address Update - When set will not update the source addresses for this port.
4	I0_REG_P0_NO_LEARN	R/W	0h	No Learn - When set will not learn the source addresses for this port.
3	I0_REG_P0_VID_INGRESS_CHECK	R/W	0h	VLAN Ingress Check - When set if a packet received is not a member of the VLAN, the packet will be dropped.
2	I0_REG_P0_DROP_UNTAGGED	R/W	0h	If Drop Untagged - When set will drop packets without a VLAN tag.
1-0	I0_REG_P0_PORTSTATE	R/W	0h	Port State - Defines the current port state used for lookup operations. 0 - Disabled 1 - Blocked 2 - Learning 3 - Forwarding

### 9.6.638 ALE\_I0\_ALE\_PORTCTL0\_1 Register (Offset = 0003E044h) [Reset = 000000X0h]

ALE\_I0\_ALE\_PORTCTL0\_1 is shown in [Table 9-670](#).

Return to the [Summary Table](#).

The ALE Port Control Register sets the port specific modes of operation.

**Table 9-670. ALE\_I0\_ALE\_PORTCTL0\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	I0_REG_P0_BCAST_LIMIT	R/W	0h	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field.
23-16	I0_REG_P0_MCAST_LIMIT	R/W	0h	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field. The ~imcast_limit is the number of Multicast packets that will be forwarded per ~iale_prescale time.
15	I0_REG_P0_DROP_DOUBLE_VLAN	R/W	0h	Drop Double VLAN - When set cause any received packet with double VLANs to be dropped. That is if there are two ctag or two stag fields in the packet it will be dropped.
14	I0_REG_P0_DROP_DUAL_VLAN	R/W	0h	Drop Dual VLAN - When set will cause any received packet with dual VLAN stag followed by ctag to be dropped.
13	I0_REG_P0_MACONLY_CAF	R/W	0h	Mac Only Copy All Frames - When set a Mac Only port will transfer all received good frames to the host. When clear a Mac Only port will transfer packets to the host based on ALE destination address lookup operation (which operates more like an Ethernet Mac). A Mac Only port is a port with ~imaconly set.
12	I0_REG_P0_DISABLE_PAUTH_MOD	R/W	0h	Disable Port authorization - When set will allow unknown addresses to arrive on a switch in authorization mode. It is intended for device to device network connection on ports which do not require MACSEC encryption.
11	I0_REG_P0_MACONLY	R/W	0h	MAC Only - When set enables this port be treated like a MAC port for the host. All traffic received is only sent to the host. The host must direct traffic to this port as the lookup engine will not send traffic to the ports with the ~ip0_maconly bit set and the ~ip0_no_learn also set. If ~ip0_maconly bit is set and the ~ip0_no_learn is not set, the host can send non-directed packets that can be sent to the destination of a MacOnly port. It is also possible that The host can broadcast to all ports including MacOnly ports in this mode.
10	I0_REG_P0_TRUNKEN	R/W	0h	Trunk Enable - This field is used to enable a port into a trunk. Any port can be used as a trunk port, any two or more ports with the ~ip0_trunken its set and having the same ~ip0_trunknum will be placed in the same trunk. There is no requirement for trunk ports to be adjacent. If all ports are enabled in the same trunk, no traffic can flow as traffic received within a trunk is never trasnmitted out the same trunk. If only a single port is a member of a trunk, it looks like a normal port with exception of entries in the look up table will be noted as a trunk entry.



**Table 9-670. ALE\_I0\_ALE\_PORTCTL0\_1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9-8	I0_REG_P0_TRUNKNUM	R/W	0h	Trunk Number - This field is used as the trunk number when the ~ip0_trunken is also set. Ports with the same trunk number that have the ~ip0_trunken also set will have traffic distributed within the trunk based on the result of the hash function described above.
7	I0_REG_P0_MIRROR_SP	R/W	0h	Mirror Source Port - This field enables the source port mirror option. When this bit is set any traffic received on the port with the reg_p0_mirror_sp bit set will have its received traffic also sent to the ~imirror_top port.
6	RESERVED	R	0h	
5	I0_REG_P0_NO_SA_UPDATE	R/W	0h	No Source Address Update - When set will not update the source addresses for this port.
4	I0_REG_P0_NO_LEARN	R/W	0h	No Learn - When set will not learn the source addresses for this port.
3	I0_REG_P0_VID_INGRESS_CHECK	R/W	0h	VLAN Ingress Check - When set if a packet received is not a member of the VLAN, the packet will be dropped.
2	I0_REG_P0_DROP_UNTAGGED	R/W	0h	If Drop Untagged - When set will drop packets without a VLAN tag.
1-0	I0_REG_P0_PORTSTATE	R/W	0h	Port State - Defines the current port state used for lookup operations. 0 - Disabled 1 - Blocked 2 - Learning 3 - Forwarding

### 9.6.639 ALE\_ALE\_UVLAN\_MEMBER Register (Offset = 0003E090h) [Reset = 00000000h]

ALE\_ALE\_UVLAN\_MEMBER is shown in [Table 9-671](#).

Return to the [Summary Table](#).

The ALE Unknown VLAN Member Mask Register is used to specify the member list for unknown VLAN ID.

**Table 9-671. ALE\_ALE\_UVLAN\_MEMBER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	UVLAN_MEMBER_LIST	R/W	0h	Unknown VLAN Member List - Each bit represents the port member status for unknown VLANs.

### 9.6.640 ALE\_ALE\_UVLAN\_URCAST Register (Offset = 0003E094h) [Reset = 00000000h]

ALE\_ALE\_UVLAN\_URCAST is shown in [Table 9-672](#).

Return to the [Summary Table](#).

The ALE Unknown VLAN Unregistered Multicast Flood Mask Register is used to specify which egress ports unregistered multicast addresses egress for the unregistered VLAN ID.

**Table 9-672. ALE\_ALE\_UVLAN\_URCAST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	UVLAN_UNREG_MCAST_FLOOD_MASK	R/W	0h	Unknown VLAN Unregister Multicast Flood Mask - Each bit represents the port to which unregistered multicast are sent for unregistered VLANs.

### 9.6.641 ALE\_ALE\_UVLAN\_RMCAST Register (Offset = 0003E098h) [Reset = 00000000h]

ALE\_ALE\_UVLAN\_RMCAST is shown in [Table 9-673](#).

Return to the [Summary Table](#).

The ALE Unknown VLAN Registered Multicast Flood Mask Register is used to specify which egress ports registered multicast addresses egress for the unregistered VLAN ID.

**Table 9-673. ALE\_ALE\_UVLAN\_RMCAST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	UVLAN_REG_MCAST_FLOOD_MASK	R/W	0h	Unknown VLAN Register Multicast Flood Mask - Each bit represents the port to which registered multicast are sent for unregistered VLANs. This field is ANDed with the registered multicast mask to determine the destinations for unregistered VLANs.

### 9.6.642 ALE\_ALE\_UVLAN\_UNTAG Register (Offset = 0003E09Ch) [Reset = 00000000h]

ALE\_ALE\_UVLAN\_UNTAG is shown in [Table 9-674](#).

Return to the [Summary Table](#).

The ALE Unknown VLAN force Untagged Egress Mask Register is used to specify which egress ports the VLAN ID will be removed.

**Table 9-674. ALE\_ALE\_UVLAN\_UNTAG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	UVLAN_FORCE_UNTAGGED_EGRESS	R/W	0h	Unknown VLAN Force Untagged Egress Mask - Each bit represents the port where the VLAN will be removed for unregistered VLANs.

### 9.6.643 ALE\_ALE\_FAST\_LUT Register (Offset = 0003E0B4h) [Reset = 00000000h]

ALE\_ALE\_FAST\_LUT is shown in [Table 9-675](#).

Return to the [Summary Table](#).

The Fast LUT registers allows the ports to be placed in Fast LUT mode.

**Table 9-675. ALE\_ALE\_FAST\_LUT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	FAST_LUT	R/W	0h	The ~Fast_LUT field allows any port to be Fast_LUT mode, which will cause all lookup operations to start based on DA/SA and VLAN only. That is any data beyond the first 32 are not used in the lookup process.

### 9.6.644 ALE\_ALE\_STAT\_DIAG Register (Offset = 0003E0B8h) [Reset = 0000XXX0h]

ALE\_ALE\_STAT\_DIAG is shown in [Table 9-676](#).

Return to the [Summary Table](#).

The ALE Statistic Output Diagnostic Register allows the output statistics to diagnose the SW counters. This register is for diagnostic only.

**Table 9-676. ALE\_ALE\_STAT\_DIAG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	PBCAST_DIAG	R/W	0h	When set and the ~ipport_diag is set to zero, will allow all ports to see the same stat diagnostic increment.
14-9	RESERVED	R	0h	
8	PORT_DIAG	R/W	0h	The port selected that a received packet will cause the selected error to increment
7-4	RESERVED	R	0h	
3-0	STAT_DIAG	R/W	0h	When non-zero will cause the selected statistic to increment on the next frame received. For the selected Port. 0: Disabled 1: Destination Equal Source Drop Stat will count 2: VLAN Ingress Check Drop Stat will count 3: Source Multicast Drop Stat will count 4: Dual VLAN Drop Stat will count 5: Ether Type length error Drop Stat will count 6: Next Hop Limit Drop Stat will count 7: IPv4 Fragment Drop Stat will count 8: Classifier Hit Stat will count 9: Classifier Red Drop Stat will count 10: Classifier Yellow Drop Stat will count 11: ALE Overflow Drop Stat will count 12: Rate Limit Drop Stat will count 13: Blocked Address Drop Stat will count 14: Secure Address Drop Stat will count 15: Authorization Drop Stat will count.

### 9.6.645 ALE\_ALE\_OAM\_LB\_CTRL Register (Offset = 0003E0BCh) [Reset = 00000000h]

ALE\_ALE\_OAM\_LB\_CTRL is shown in [Table 9-677](#).

Return to the [Summary Table](#).

The ALE OAM Control allows ports to be put into OAM Loopback, only non-supervisor packet are looped back to the source port.

**Table 9-677. ALE\_ALE\_OAM\_LB\_CTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	OAM_LB_CTRL	R/W	0h	The ~ioam_lb_ctrl allows any port to be put into OAM loopback, that is any packet received will be returned to the same port with an egressop of 0xFF which swaps the source and destination address. BPDUs will still flow through as normal so that OAM can be remotely requested and disabled.



### 9.6.646 ALE\_ALE\_MSK\_MUX0 Register (Offset = 0003E0C0h) [Reset = 00000003h]

ALE\_ALE\_MSK\_MUX0 is shown in [Table 9-678](#).

Return to the [Summary Table](#).

VLAN Mask Mux x - The ALE Mask Mux registers are used along with the VLAN registered/unregistered index selectors from the Lookup Table to determine the value for vlan registered and unregister mask respectively.

**Table 9-678. ALE\_ALE\_MSK\_MUX0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	VLAN_MASK_MUX_0	R	3h	VLAN Mask Mux x - When selected by the VLAN lookup table entry FwdUnRegIdx or FwdAllRegIdx is used as the FwdUnRegMask or FwdUnRegMask values anded with the member list to determine the forwarding of packets. The Value of vlan_mask_mux_0 is read only and set to all ones for all ports.

### 9.6.647 ALE\_I1\_ALE\_MSK\_MUX1\_0 Register (Offset = 0003E0C4h) [Reset = 00000000h]

ALE\_I1\_ALE\_MSK\_MUX1\_0 is shown in [Table 9-679](#).

Return to the [Summary Table](#).

VLAN Mask Mux x - The ALE Mask Mux registers are used along with the VLAN registered/unregistered index selectors from the Lookup Table to determine the value for vlan registered and unregister mask respectively.

**Table 9-679. ALE\_I1\_ALE\_MSK\_MUX1\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	I1_REG_VLAN_MASK_MUX_1	R/W	0h	VLAN Mask Mux x - When selected by the VLAN lookup table entry FwdUnRegIdx or FwdAllRegIdx is used as the FwdUnRegMask or FwdUnRegMask values anded with the member list to determine the forwarding of packets. The Value of vlan_mask_mux_0 is read only and set to all ones for all ports.

### 9.6.648 ALE\_I1\_ALE\_MSK\_MUX1\_1 Register (Offset = 0003E0C8h) [Reset = 00000000h]

ALE\_I1\_ALE\_MSK\_MUX1\_1 is shown in [Table 9-680](#).

Return to the [Summary Table](#).

VLAN Mask Mux x - The ALE Mask Mux registers are used along with the VLAN registered/unregistered index selectors from the Lookup Table to determine the value for vlan registered and unregister mask respectively.

**Table 9-680. ALE\_I1\_ALE\_MSK\_MUX1\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	I1_REG_VLAN_MASK_MUX_1	R/W	0h	VLAN Mask Mux x - When selected by the VLAN lookup table entry FwdUnRegIdx or FwdAllRegIdx is used as the FwdUnRegMask or FwdUnRegMask values anded with the member list to determine the forwarding of packets. The Value of vlan_mask_mux_0 is read only and set to all ones for all ports.

### 9.6.649 ALE\_I1\_ALE\_MSK\_MUX1\_2 Register (Offset = 0003E0CCh) [Reset = 00000000h]

ALE\_I1\_ALE\_MSK\_MUX1\_2 is shown in [Table 9-681](#).

Return to the [Summary Table](#).

VLAN Mask Mux x - The ALE Mask Mux registers are used along with the VLAN registered/unregistered index selectors from the Lookup Table to determine the value for vlan registered and unregister mask respectively.

**Table 9-681. ALE\_I1\_ALE\_MSK\_MUX1\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	I1_REG_VLAN_MASK_MUX_1	R/W	0h	VLAN Mask Mux x - When selected by the VLAN lookup table entry FwdUnRegIdx or FwdAllRegIdx is used as the FwdUnRegMask or FwdUnRegMask values anded with the member list to determine the forwarding of packets. The Value of vlan_mask_mux_0 is read only and set to all ones for all ports.

### 9.6.650 ALE\_EGRESSOP Register (Offset = 0003E0FCh) [Reset = 000XXXXh]

ALE\_EGRESSOP is shown in [Table 9-682](#).

Return to the [Summary Table](#).

The Egress Operation register allows enabled classifiers with any match like IPSA or IPDA match to use the CPSW Egress Packet Operations Inter VLAN Routing sub functions. If the packet was destined for the host or is destined to any port without any errors, but matches a classifier that has a programmed egress opcode, it will be forwarded to the destination ports where the destination ports will use the thier egress opcode entry to modify the packet. InterVLAN Routing and mirroring need to be understood, they are orthogonal functions. Care must be taken not to violate VLAN rules as this can redirect packets based on classifier matches.

**Table 9-682. ALE\_EGRESSOP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	EGRESS_OP	R/W	0h	The Egress Operation defines the operation performed by the CPSW Egress Packet Operations 0: NOP : 1-n: Defines which egress Operation will be performed. This allows Inter VLAN routing to be configured for high bandwidth traffic, reducing CPU load. 0xff: Swap SA and DA of packet, this is intended to allow OAM diagnostics for a link.
23-21	EGRESS_TRK	R/W	0h	The Egress Trunk Index is the calculated trunk index from the SA, DA or VLAN if modified to that InterVLAN routing will work on trunks as well. The DA, SA and VLAN are ignored for trunk generation on InterVLAN Routing so that this field is the index generated from the Egress Op replacements elcusive or'd together into a three bit index.
20	TTL_CHECK	R/W	0h	The TTL Check will cause any packet that fails TTL checks to not be routed to the Inter VLAN Routing sub functions. The packet will be routed to the host it was destined to.
19-2	RESERVED	R	0h	
1-0	DEST_PORTS	R/W	0h	The Destination Ports is a list of the ports the classified packet will be set to. If a destination is a Trunk, all the port bits for that trunck must be set.

### 9.6.651 ALE\_POLICECFG0 Register (Offset = 0003E100h) [Reset = XXX0XXX0h]

ALE\_POLICECFG0 is shown in [Table 9-683](#).

Return to the [Summary Table](#).

The Policing Config 0 holds the port, frame priority and ONU address index as well as match enables for port, frame priority and ONU address matching.

**Table 9-683. ALE\_POLICECFG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PORT_MEN	R/W	0h	Port Match Enable - Enabled port match for the selected policing/classifier entry
30	TRUNKID	R/W	0h	Trunk ID - When set indicates the port number is a trunk group.
29-26	RESERVED	R	0h	
25	PORT_NUM	R/W	0h	Port Number - Specifies the port address to match for the selected policing/classifier entry
24-20	RESERVED	R	0h	
19	PRI_MEN	R/W	0h	Priority Match Enable - Enables frame priority match for the selected policing/classifier entry
18-16	PRI_VAL	R/W	0h	Priority Value - Specifies the frame priority to match for the selected policing/classifier entry
15	ONU_MEN	R/W	0h	OUI Match Enable - Enables frame ONU address match for the selected policing/classifier entry
14-5	RESERVED	R	0h	
4-0	ONU_INDEX	R/W	0h	OUI Table Entry Index - Specifies the ALE ONU address lookup table index to match for the selected policing/classifier entry

### 9.6.652 ALE\_POLICECFG1 Register (Offset = 0003E104h) [Reset = XXX0XXX0h]

ALE\_POLICECFG1 is shown in [Table 9-684](#).

Return to the [Summary Table](#).

The Policing Config 1 holds the match enable/match index for the L2 Destination and L2 source addresses

**Table 9-684. ALE\_POLICECFG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	DST_MEN	R/W	0h	Destination Address Match Enable - Enables frame L2 destination address match for the selected policing/classifier entry
30-21	RESERVED	R	0h	
20-16	DST_INDEX	R/W	0h	Destination Address Table Entry Index - Specifies the ALE L2 destination address lookup table index to match for the selected policing/classifier entry
15	SRC_MEN	R/W	0h	Source Address Match Enable - Enables frame L2 source address match for the selected policing/classifier entry
14-5	RESERVED	R	0h	
4-0	SRC_INDEX	R/W	0h	Source Address Table Entry Index - Specifies the ALE L2 source address lookup table index to match for the selected policing/classifier entry

### 9.6.653 ALE\_POLICECFG2 Register (Offset = 0003E108h) [Reset = XXX0XXX0h]

ALE\_POLICECFG2 is shown in [Table 9-685](#).

Return to the [Summary Table](#).

The Policing Config 2 holds the match enable/match index for the Outer VLAN and Inner VLAN addresses

**Table 9-685. ALE\_POLICECFG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	OVLAN_MEN	R/W	0h	Outer VLAN Match Enable - Enables frame Outer VLAN address match for the selected policing/classifier entry
30-21	RESERVED	R	0h	
20-16	OVLAN_INDEX	R/W	0h	Outer VLAN Table Entry Index - Specifies the ALE Outer VLAN address lookup table index to match for the selected policing/classifier entry Note this index assumes the VLANID is in the packet, it does not use the port VLAN if the packet is untagged or priority tagged.
15	IVLAN_MEN	R/W	0h	Inner VLAN Match Enable - Enables frame Inner VLAN address match for the selected policing/classifier entry
14-5	RESERVED	R	0h	
4-0	IVLAN_INDEX	R/W	0h	Inner VLAN Table Entry Index - Specifies the ALE Inner VLAN address lookup table index to match for the selected policing/classifier entry Note this index assumes the VLANID is in the packet, it does not use the port VLAN if the packet is untagged or priority tagged.



### 9.6.654 ALE\_POLICECFG3 Register (Offset = 0003E10Ch) [Reset = XXX0XXX0h]

ALE\_POLICECFG3 is shown in [Table 9-686](#).

Return to the [Summary Table](#).

The Policing Config 3 holds the match enable/match index for the Ether Type and IP Source address

**Table 9-686. ALE\_POLICECFG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	ETHERTYPE_MEN	R/W	0h	EtherType Match Enable - Enables frame Ether Type match for the selected policing/classifier entry
30-21	RESERVED	R	0h	
20-16	ETHERTYPE_INDEX	R/W	0h	EtherType Table Entry Index - Specifies the ALE Ether Type lookup table index to match for the selected policing/classifier entry
15	IPSRC_MEN	R/W	0h	IP Source Address Match Enable - Enables frame IP Source address match for the selected policing/classifier entry
14-5	RESERVED	R	0h	
4-0	IPSRC_INDEX	R/W	0h	IP Source Address Table Entry Index - Specifies the ALE IP Source address lookup table index to match for the selected policing/classifier entry

### 9.6.655 ALE\_POLICECFG4 Register (Offset = 0003E110h) [Reset = XXX0XXXXh]

ALE\_POLICECFG4 is shown in [Table 9-687](#).

Return to the [Summary Table](#).

The Policing Config 4 holds the match enable/match index for the IP Destination address

**Table 9-687. ALE\_POLICECFG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	IPDST_MEN	R/W	0h	IP Destination Address Match Enable - Enables frame IP Destination address match for the selected policing/classifier entry
30-21	RESERVED	R	0h	
20-16	IPDST_INDEX	R/W	0h	IP Destination Address Table Entry Index - Specifies the ALE IP Destination address lookup table index to match for the selected policing/classifier entry
15-0	RESERVED	R	0h	

### 9.6.656 ALE\_POLICECFG6 Register (Offset = 0003E118h) [Reset = 00000000h]

ALE\_POLICECFG6 is shown in [Table 9-688](#).

Return to the [Summary Table](#).

The PIR counter is a 37 bit internal counter where ~ipir\_idle\_inc\_val is added every clock and the frame size << 18 is subtracted at EOF if not RED at LUT time. If the counter is negative the packet will be marked RED, else it can be YELLOW or GREEN based on the CIR counter. If only this counter is used (aka cir\_idle\_inc\_val==0) Packet are marked RED or GREEN based on PIR counter only.

**Table 9-688. ALE\_POLICECFG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PIR_IDLE_INC_VAL	R/W	0h	Peak Information Rate Idle Increment Value - The number added to the PIR counter every clock cycle. If zero the PIR counter is disabled and packets will never be marked or processed as RED.

### 9.6.657 ALE\_POLICECFG7 Register (Offset = 0003E11Ch) [Reset = 00000000h]

ALE\_POLICECFG7 is shown in [Table 9-689](#).

Return to the [Summary Table](#).

The CIR counter is a 37 bit internal counter where ~icir\_idle\_inc\_val is added every clock and the frame size &lt;&lt; 18 is subtracted at EOF if not RED or YELLOW at LUT time. If the counter is positive the packet will be marked GREEN, else it can be YELLOW or RED based on the PIR counter. If only this counter is used (aka pir\_idle\_inc\_val==0) Packet are marked YELLOW or GREEN based on CIR counter only.

**Table 9-689. ALE\_POLICECFG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CIR_IDLE_INC_VAL	R/W	0h	Committed Information Idle Increment Value - The number added to the CIR counter every clock cycle. If zero the CIR counter is disabled and packets will never be marked or processed as YELLOW.

### 9.6.658 ALE\_POLICETBLCTL Register (Offset = 0003E120h) [Reset = XXXXXXXXh]

ALE\_POLICETBLCTL is shown in [Table 9-690](#).

Return to the [Summary Table](#).

The Policing Table Control is used to read or write the selected policing/classifier entry. The selected policing/classifier entry is only read or written after this register is written based on the value of the ~iwrite\_enable bit.

**Table 9-690. ALE\_POLICETBLCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	WRITE_ENABLE	R/W	0h	Write Enable - Setting this bit will write the POLICECFG0-7 to the ~ipol_tbl_idx selected policing/classifier entry. Clearing this bit will read the ~ipol_tbl_idx selected policing/classifier entry into the POLICECFG0-7 registers.
30-2	RESERVED	R	0h	
1-0	POL_TBL_IDX	R/W	0h	<p>Policer Entry Index - This field specifies the policing/classifier entry to be read or written.</p> <p>When writing to this field without setting the ~iwrite_enable=1 will cause the selected policing/classifier entry to be loaded into the POLICECFG0-7 registers.</p> <p>When writing to this field with setting the ~iwrite_enable=1 will cause the selected policing/classifier entry to be updated from the POLICECFG0-7 registers.</p>

### 9.6.659 ALE\_POLICECONTROL Register (Offset = 0003E124h) [Reset = XX0XXXXXh]

ALE\_POLICECONTROL is shown in [Table 9-691](#).

Return to the [Summary Table](#).

The Control Enables color marking as well as internal ALE packet dropping rules.

**Table 9-691. ALE\_POLICECONTROL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	POLICING_EN	R/W	0h	Policing Enable - Enables the policing to color the packets, this also enables red or yellow drop capabilities.
30	RESERVED	R	0h	
29	RED_DROP_EN	R/W	0h	RED Drop Enable - Enables the ALE to drop the red colored packets.
28	YELLOW_DROP_EN	R/W	0h	YELLOW Drop Enable - Enables the ALE to drop yellow packets based on the ~iyellowthresh value. This field would normally not be used as to let the switch drop packets at a buffer threshold instead. In the event that the switch does not enable buffer threshold dropping, YELLOW packets can be dropped based on this feature.
27	RESERVED	R	0h	
26-24	YELLOWTHRESH	R/W	0h	Yellow Threshold - When set enables a portion of the yellow packets to be dropped based on the ~iyellow_drop_en enable. 0- 100% 1= 50% 2- 33% 3- 25% 4= 20% 5- 17% 6- 14% 7- 13%
23-22	POLMCHMODE	R/W	0h	Policing Match Mode - This field determines what happens to packets that fail to hit any policing/classifier entry. 0 - No Hit packets are marked GREEN 1 - No Hit packets are marked YELLOW 2 - No Hit packets are marked RED 3 - No Hit packets are marked based on policing/classifier entry=0 state.
21	PRIORITY_THREAD_EN	R/W	0h	Priority Thread Enable - This field determines if priority is OR'd to the default thread when no classifiers hit and the default thread is enabled.
20	MAC_ONLY_DEF_DIS	R/W	0h	MAC Only Default Disable - This field when set disables the default thread on MAC Only Ports. That is the default thread will be {port,priority}. If the traffic matches a classifier with a thread mapping, the classifier thread mapping still occurs.
19-0	RESERVED	R	0h	

### 9.6.660 ALE\_POLICETESTCTL Register (Offset = 0003E128h) [Reset = 0XXXXXXXh]

ALE\_POLICETESTCTL is shown in [Table 9-692](#).

Return to the [Summary Table](#).

The Policing Test Control enables the ability to determine which policing entry has been hit and whether they reported a red or yellow rate condition.

**Table 9-692. ALE\_POLICETESTCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	POL_CLRALL_HIT	R/W	0h	Policer Clear - This bit clears all the policing/classifier hit bits. This bit is self clearing. This can be used to test the fact that a policing/classifier entry has been hit.
30	POL_CLRALL_REDHIT	R/W	0h	Policer Clear RED - This bit clears all the policing/classifier RED hit bits. This bit is self clearing. This can be used to test the fact that a policing/classifier entry has been hit during a RED condition.
29	POL_CLRALL_YELLOWHIT	R/W	0h	Policer Clear YELLOW - This bit clears all the policing/classifier YELLOW hit bits. This bit is self clearing. This can be used to test the fact that a policing/classifier entry has been hit during a YELLOW condition.
28	POL_CLRSEL_ALL	R/W	0h	Police Clear Selected - This bit clears the selected policing/classifier hit, redhit and yellowhit bits. This bit is self clearing.
27-2	RESERVED	R	0h	
1-0	POL_TEST_IDX	R/W	0h	Policer Test Index - This field selects which policing/classifier hit bits will be read or written.

### 9.6.661 ALE\_POLICEHSTAT Register (Offset = 0003E12Ch) [Reset = XXXXXXXXh]

ALE\_POLICEHSTAT is shown in [Table 9-693](#).

Return to the [Summary Table](#).

The policing hit status is a read only register that reads the hit bits of the selected policing/classifier.

**Table 9-693. ALE\_POLICEHSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	POL_HIT	R	0h	Policer Hit - This indicates that the selected policing/classifier via the ~ipol_test_idx field has been hit by a packet seen on any port that matches the policing/classifier entry match.
30	POL_REDHIT	R	0h	Policer Hit RED - This indicates that the selected policing/classifier via the ~ipol_test_idx field has been hit during a RED condition by a packet seen on any port that matches the policing/classifier entry match.
29	POL_YELLOWHIT	R	0h	Policer Hit YELLOW - This indicates that the selected policing/classifier via the ~ipol_test_idx field has been hit during a YELLOW condition by a packet seen on any port that matches the policing/classifier entry match.
28-0	RESERVED	R	0h	



### 9.6.662 ALE\_THREADMAPDEF Register (Offset = 0003E134h) [Reset = 0000XXX0h]

ALE\_THREADMAPDEF is shown in [Table 9-694](#).

Return to the [Summary Table](#).

The THREAD Mapping Default Value register is used to set the default thread ID when no classifier is matched,

**Table 9-694. ALE\_THREADMAPDEF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	DEFTHREAD_EN	R/W	0h	Default Tread Enable - When set the switch will use the ~idefthreadval for the host interface thread ID if no classifier is matched. If clear the switch will generate its own thread ID based on port and priority if there is no classifier match.
14-6	RESERVED	R	0h	
5-0	DEFTHREADVAL	R/W	0h	Default Thread Value - This field specifies the default thread ID value.

### 9.6.663 ALE\_THREADMAPCTL Register (Offset = 0003E138h) [Reset = 00000000h]

ALE\_THREADMAPCTL is shown in [Table 9-695](#).

Return to the [Summary Table](#).

The THREAD Mapping Control register allows the highest matched classifier to return a particular thread ID for traffic sent to the host. This allows particular classifier matched traffic to be placed on a particular host's queue.

**Table 9-695. ALE\_THREADMAPCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1-0	CLASSINDEX	R/W	0h	Classifier Index - This is the classifier index entry that the thread enable and thread value will be read or written by the ~bTHREADMAPVAL register.

### 9.6.664 ALE\_THREADMAPVAL Register (Offset = 0003E13Ch) [Reset = 0000XXX0h]

ALE\_THREADMAPVAL is shown in [Table 9-696](#).

Return to the [Summary Table](#).

The THREAD Mapping Value register is used to set the thread ID for a particular classifier entry.

**Table 9-696. ALE\_THREADMAPVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15	THREAD_EN	R/W	0h	Thread Enable - When set the switch will use the ~ithreadval for the selected classifier match. If clear the the thread ID will be determined by the ~bTHREADMAPDEF register settings.
14-6	RESERVED	R	0h	
5-0	THREADVAL	R/W	0h	Thread Value - This field is the thread ID value that is used to map a classifier hit to thread ID for host traffic.

### 9.6.665 rev Register (Offset = 0003F000h) [Reset = 66A03A01h]

rev is shown in [Table 9-697](#).

Return to the [Summary Table](#).

Revision parameters

**Table 9-697. rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A0h	Module ID
15-11	revrtl	R	7h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	1h	Minor version

### 9.6.666 vector Register (Offset = 0003F008h) [Reset = 0000XX00h]

vector is shown in [Table 9-698](#).

Return to the [Summary Table](#).

ECC Vector Register

**Table 9-698. vector Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	rd_svbus_done	R/W1C	0h	Status to indicate if read on serial VBUS is complete, write of any value will clear this bit.
23-16	rd_svbus_address	R/W	0h	Read address
15	rd_svbus	R/W1S	0h	Write 1 to trigger a read on the serial VBUS
14-11	RESERVED	R	0h	
10-0	ecc_vector	R/W	0h	Value written to select the corresponding ECC RAM for control or status

### 9.6.667 stat Register (Offset = 0003F00Ch) [Reset = 00000005h]

stat is shown in [Table 9-699](#).

Return to the [Summary Table](#).

Misc Status

**Table 9-699. stat Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	0h	
10-0	num_rams	R	5h	Indicates the number of RAMS serviced by the ECC aggregator

### 9.6.668 ECC\_reserved\_svbus\_0 Register (Offset = 0003F010h) [Reset = 00000000h]

ECC\_reserved\_svbus\_0 is shown in [Table 9-700](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-700. ECC\_reserved\_svbus\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.669 ECC\_reserved\_svbus\_1 Register (Offset = 0003F014h) [Reset = 00000000h]

ECC\_reserved\_svbus\_1 is shown in [Table 9-701](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-701. ECC\_reserved\_svbus\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data



### 9.6.670 ECC\_reserved\_svbus\_2 Register (Offset = 0003F018h) [Reset = 00000000h]

ECC\_reserved\_svbus\_2 is shown in [Table 9-702](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-702. ECC\_reserved\_svbus\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.671 ECC\_reserved\_svbus\_3 Register (Offset = 0003F01Ch) [Reset = 00000000h]

ECC\_reserved\_svbus\_3 is shown in [Table 9-703](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-703. ECC\_reserved\_svbus\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.672 ECC\_reserved\_svbus\_4 Register (Offset = 0003F020h) [Reset = 00000000h]

ECC\_reserved\_svbus\_4 is shown in [Table 9-704](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-704. ECC\_reserved\_svbus\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.673 ECC\_reserved\_svbus\_5 Register (Offset = 0003F024h) [Reset = 00000000h]

ECC\_reserved\_svbus\_5 is shown in [Table 9-705](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-705. ECC\_reserved\_svbus\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.674 ECC\_reserved\_svbus\_6 Register (Offset = 0003F028h) [Reset = 00000000h]

ECC\_reserved\_svbus\_6 is shown in [Table 9-706](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-706. ECC\_reserved\_svbus\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.675 ECC\_reserved\_svbus\_7 Register (Offset = 0003F02Ch) [Reset = 00000000h]

ECC\_reserved\_svbus\_7 is shown in [Table 9-707](#).

Return to the [Summary Table](#).

Reference other documents that contain the ECC RAM wrapper and EDC controller serial vbus register sets.

**Table 9-707. ECC\_reserved\_svbus\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	data	R/W	0h	Serial VBUS register data

### 9.6.676 ECC\_sec\_eoi\_reg Register (Offset = 0003F03Ch) [Reset = 00000000h]

ECC\_sec\_eoi\_reg is shown in [Table 9-708](#).

Return to the [Summary Table](#).

EOI Register

**Table 9-708. ECC\_sec\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	eoi_wr	R/W1S	0h	EOI Register

### 9.6.677 ECC\_sec\_status\_reg0 Register (Offset = 0003F040h) [Reset = 00000000h]

ECC\_sec\_status\_reg0 is shown in [Table 9-709](#).

Return to the [Summary Table](#).

Interrupt Status Register 0

**Table 9-709. ECC\_sec\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	ramecc19_pend	R/W1S	0h	Interrupt Pending Status for ramecc19_pend
18	ramecc18_pend	R/W1S	0h	Interrupt Pending Status for ramecc18_pend
17	ramecc17_pend	R/W1S	0h	Interrupt Pending Status for ramecc17_pend
16	ramecc16_pend	R/W1S	0h	Interrupt Pending Status for ramecc16_pend
15	ramecc15_pend	R/W1S	0h	Interrupt Pending Status for ramecc15_pend
14	ramecc14_pend	R/W1S	0h	Interrupt Pending Status for ramecc14_pend
13	ramecc13_pend	R/W1S	0h	Interrupt Pending Status for ramecc13_pend
12	ramecc12_pend	R/W1S	0h	Interrupt Pending Status for ramecc12_pend
11	ramecc11_pend	R/W1S	0h	Interrupt Pending Status for ramecc11_pend
10	ramecc10_pend	R/W1S	0h	Interrupt Pending Status for ramecc10_pend
9	ramecc9_pend	R/W1S	0h	Interrupt Pending Status for ramecc9_pend
8	ramecc8_pend	R/W1S	0h	Interrupt Pending Status for ramecc8_pend
7	ramecc7_pend	R/W1S	0h	Interrupt Pending Status for ramecc7_pend
6	ramecc6_pend	R/W1S	0h	Interrupt Pending Status for ramecc6_pend
5	ramecc5_pend	R/W1S	0h	Interrupt Pending Status for ramecc5_pend
4	ramecc4_pend	R/W1S	0h	Interrupt Pending Status for ramecc4_pend
3	ramecc3_pend	R/W1S	0h	Interrupt Pending Status for ramecc3_pend
2	ramecc2_pend	R/W1S	0h	Interrupt Pending Status for ramecc2_pend
1	ramecc1_pend	R/W1S	0h	Interrupt Pending Status for ramecc1_pend
0	ramecc0_pend	R/W1S	0h	Interrupt Pending Status for ramecc0_pend



### 9.6.678 ECC\_sec\_enable\_set\_reg0 Register (Offset = 0003F080h) [Reset = 00000000h]

ECC\_sec\_enable\_set\_reg0 is shown in [Table 9-710](#).

Return to the [Summary Table](#).

Interrupt Enable Set Register 0

**Table 9-710. ECC\_sec\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	ramecc19_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc19_pend
18	ramecc18_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc18_pend
17	ramecc17_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc17_pend
16	ramecc16_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc16_pend
15	ramecc15_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc15_pend
14	ramecc14_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc14_pend
13	ramecc13_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc13_pend
12	ramecc12_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc12_pend
11	ramecc11_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc11_pend
10	ramecc10_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc10_pend
9	ramecc9_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc9_pend
8	ramecc8_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc8_pend
7	ramecc7_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc7_pend
6	ramecc6_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc6_pend
5	ramecc5_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc5_pend
4	ramecc4_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc4_pend
3	ramecc3_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc3_pend
2	ramecc2_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc2_pend
1	ramecc1_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc1_pend
0	ramecc0_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc0_pend

### 9.6.679 ECC\_sec\_enable\_clr\_reg0 Register (Offset = 0003F0C0h) [Reset = 00000000h]

ECC\_sec\_enable\_clr\_reg0 is shown in [Table 9-711](#).

Return to the [Summary Table](#).

Interrupt Enable Clear Register 0

**Table 9-711. ECC\_sec\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	ramecc19_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc19_pend
18	ramecc18_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc18_pend
17	ramecc17_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc17_pend
16	ramecc16_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc16_pend
15	ramecc15_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc15_pend
14	ramecc14_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc14_pend
13	ramecc13_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc13_pend
12	ramecc12_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc12_pend
11	ramecc11_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc11_pend
10	ramecc10_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc10_pend
9	ramecc9_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc9_pend
8	ramecc8_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc8_pend
7	ramecc7_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc7_pend
6	ramecc6_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc6_pend
5	ramecc5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc5_pend
4	ramecc4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc4_pend
3	ramecc3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc3_pend
2	ramecc2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc2_pend
1	ramecc1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc1_pend
0	ramecc0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc0_pend

### 9.6.680 ECC\_ded\_eoi\_reg Register (Offset = 0003F13Ch) [Reset = 00000000h]

ECC\_ded\_eoi\_reg is shown in [Table 9-712](#).

Return to the [Summary Table](#).

EOI Register

**Table 9-712. ECC\_ded\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	eoi_wr	R/W1S	0h	EOI Register

### 9.6.681 ECC\_ded\_status\_reg0 Register (Offset = 0003F140h) [Reset = 00000000h]

ECC\_ded\_status\_reg0 is shown in [Table 9-713](#).

Return to the [Summary Table](#).

Interrupt Status Register 0

**Table 9-713. ECC\_ded\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	ramecc19_pend	R/W1S	0h	Interrupt Pending Status for ramecc19_pend
18	ramecc18_pend	R/W1S	0h	Interrupt Pending Status for ramecc18_pend
17	ramecc17_pend	R/W1S	0h	Interrupt Pending Status for ramecc17_pend
16	ramecc16_pend	R/W1S	0h	Interrupt Pending Status for ramecc16_pend
15	ramecc15_pend	R/W1S	0h	Interrupt Pending Status for ramecc15_pend
14	ramecc14_pend	R/W1S	0h	Interrupt Pending Status for ramecc14_pend
13	ramecc13_pend	R/W1S	0h	Interrupt Pending Status for ramecc13_pend
12	ramecc12_pend	R/W1S	0h	Interrupt Pending Status for ramecc12_pend
11	ramecc11_pend	R/W1S	0h	Interrupt Pending Status for ramecc11_pend
10	ramecc10_pend	R/W1S	0h	Interrupt Pending Status for ramecc10_pend
9	ramecc9_pend	R/W1S	0h	Interrupt Pending Status for ramecc9_pend
8	ramecc8_pend	R/W1S	0h	Interrupt Pending Status for ramecc8_pend
7	ramecc7_pend	R/W1S	0h	Interrupt Pending Status for ramecc7_pend
6	ramecc6_pend	R/W1S	0h	Interrupt Pending Status for ramecc6_pend
5	ramecc5_pend	R/W1S	0h	Interrupt Pending Status for ramecc5_pend
4	ramecc4_pend	R/W1S	0h	Interrupt Pending Status for ramecc4_pend
3	ramecc3_pend	R/W1S	0h	Interrupt Pending Status for ramecc3_pend
2	ramecc2_pend	R/W1S	0h	Interrupt Pending Status for ramecc2_pend
1	ramecc1_pend	R/W1S	0h	Interrupt Pending Status for ramecc1_pend
0	ramecc0_pend	R/W1S	0h	Interrupt Pending Status for ramecc0_pend

### 9.6.682 ECC\_ded\_enable\_set\_reg0 Register (Offset = 0003F180h) [Reset = 00000000h]

ECC\_ded\_enable\_set\_reg0 is shown in [Table 9-714](#).

Return to the [Summary Table](#).

Interrupt Enable Set Register 0

**Table 9-714. ECC\_ded\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	ramecc19_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc19_pend
18	ramecc18_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc18_pend
17	ramecc17_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc17_pend
16	ramecc16_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc16_pend
15	ramecc15_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc15_pend
14	ramecc14_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc14_pend
13	ramecc13_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc13_pend
12	ramecc12_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc12_pend
11	ramecc11_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc11_pend
10	ramecc10_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc10_pend
9	ramecc9_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc9_pend
8	ramecc8_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc8_pend
7	ramecc7_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc7_pend
6	ramecc6_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc6_pend
5	ramecc5_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc5_pend
4	ramecc4_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc4_pend
3	ramecc3_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc3_pend
2	ramecc2_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc2_pend
1	ramecc1_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc1_pend
0	ramecc0_enable_set	R/W1S	0h	Interrupt Enable Set Register for ramecc0_pend

### 9.6.683 ECC\_ded\_enable\_clr\_reg0 Register (Offset = 0003F1C0h) [Reset = 00000000h]

ECC\_ded\_enable\_clr\_reg0 is shown in [Table 9-715](#).

Return to the [Summary Table](#).

Interrupt Enable Clear Register 0

**Table 9-715. ECC\_ded\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19	ramecc19_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc19_pend
18	ramecc18_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc18_pend
17	ramecc17_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc17_pend
16	ramecc16_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc16_pend
15	ramecc15_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc15_pend
14	ramecc14_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc14_pend
13	ramecc13_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc13_pend
12	ramecc12_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc12_pend
11	ramecc11_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc11_pend
10	ramecc10_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc10_pend
9	ramecc9_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc9_pend
8	ramecc8_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc8_pend
7	ramecc7_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc7_pend
6	ramecc6_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc6_pend
5	ramecc5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc5_pend
4	ramecc4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc4_pend
3	ramecc3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc3_pend
2	ramecc2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc2_pend
1	ramecc1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc1_pend
0	ramecc0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for ramecc0_pend

### 9.6.684 aggr\_enable\_set Register (Offset = 0003F200h) [Reset = 00000000h]

aggr\_enable\_set is shown in [Table 9-716](#).

Return to the [Summary Table](#).

AGGR interrupt enable set Register

**Table 9-716. aggr\_enable\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	timeout	R/W1S	0h	interrupt enable set for svbus timeout errors
0	parity	R/W1S	0h	interrupt enable set for parity errors

### 9.6.685 aggr\_enable\_clr Register (Offset = 0003F204h) [Reset = 00000000h]

aggr\_enable\_clr is shown in [Table 9-717](#).

Return to the [Summary Table](#).

AGGR interrupt enable clear Register

**Table 9-717. aggr\_enable\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	0h	
1	timeout	R/W1C	0h	interrupt enable clear for svbus timeout errors
0	parity	R/W1C	0h	interrupt enable clear for parity errors



### 9.6.686 aggr\_status\_set Register (Offset = 0003F208h) [Reset = 00000000h]

aggr\_status\_set is shown in [Table 9-718](#).

Return to the [Summary Table](#).

AGGR interrupt status set Register

**Table 9-718. aggr\_status\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-2	timeout	R/Wincr	0h	interrupt status set for svbus timeout errors
1-0	parity	R/Wincr	0h	interrupt status set for parity errors

### 9.6.687 aggr\_status\_clr Register (Offset = 0003F20Ch) [Reset = 00000000h]

aggr\_status\_clr is shown in [Table 9-719](#).

Return to the [Summary Table](#).

AGGR interrupt status clear Register

**Table 9-719. aggr\_status\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	
3-2	timeout	R/Wdecr	0h	interrupt status clear for svbus timeout errors
1-0	parity	R/Wdecr	0h	interrupt status clear for parity errors

Chapter 10

# Quad Serial Peripheral Interface (QSPI)

---

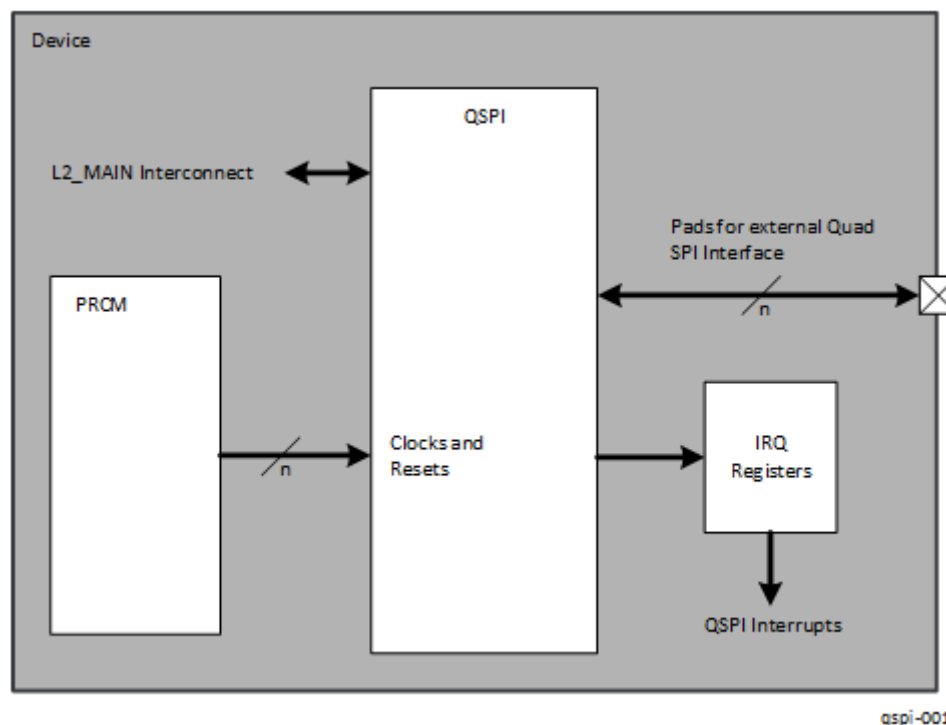


10.1 Quad Serial Peripheral Interface Overview.....	3366
10.2 QSPI Environment.....	3367
10.3 QSPI Functional Description.....	3368
10.4 QSPI Registers.....	3377

## 10.1 Quad Serial Peripheral Interface Overview

The quad serial peripheral interface (QSPI) module is a kind of SPI module that allows single, dual, or quad read access to external SPI devices. This module has a memory mapped register interface, which provides a direct interface for accessing data from external SPI devices and thus simplifying software requirements. The QSPI works as a controller only.

The one QSPI in the device is primarily intended for fast booting from quad-SPI flash memories. [Figure 10-1](#) shows the QSPI module overview.



**Figure 10-1. QSPI Overview**

The QSPI supports the following features:

- General SPI features:
  - Programmable clock divider
  - Max four pin interface
  - Programmable length (from 1 to 128 bits) of the words transferred
  - Programmable number (from 1 to 4096) of the words transferred
  - 1 external chip-select signal
  - Support for 1 pin Write. Dual or quad writes are not supported
  - Support for 1-, 2-, or 4-pin SPI interface
  - Optional interrupt generation on word or frame (number of words) completion
  - Programmable delay between chip select activation and output data from 0 to 3 QSPI clock cycles
  - Programmable signal polarities
  - Programmable active clock edge
  - Software-controllable interface allowing for any type of SPI transfer
  - Control through L2\_MAIN configuration port
- Serial flash interface (SFI) features:
  - Serial flash read/write interface
  - Additional registers for defining read and write commands to the external serial flash device
  - External flash support of up to 8 MB

- Fast read support, where fast read requires dummy bytes after address bytes; 0 to 3 dummy bytes can be configured.
- Dual read support
- Quad read support
- Little-endian support (only for memory mapped registers used to configure QSPI controller and not SPI content accesses)
- Linear increment addressing mode only

The QSPI supports only dual and quad reads. Dual or quad writes are not supported. In addition, there is no "pass through" mode supported where the data present on the QSPI input is sent to its output.

### Note

The QSPI module does not support cache line wrap mode.

## 10.2 QSPI Environment

Figure 10-2 shows a typical connection of the QSPI module to the external quad-SPI flash memory.

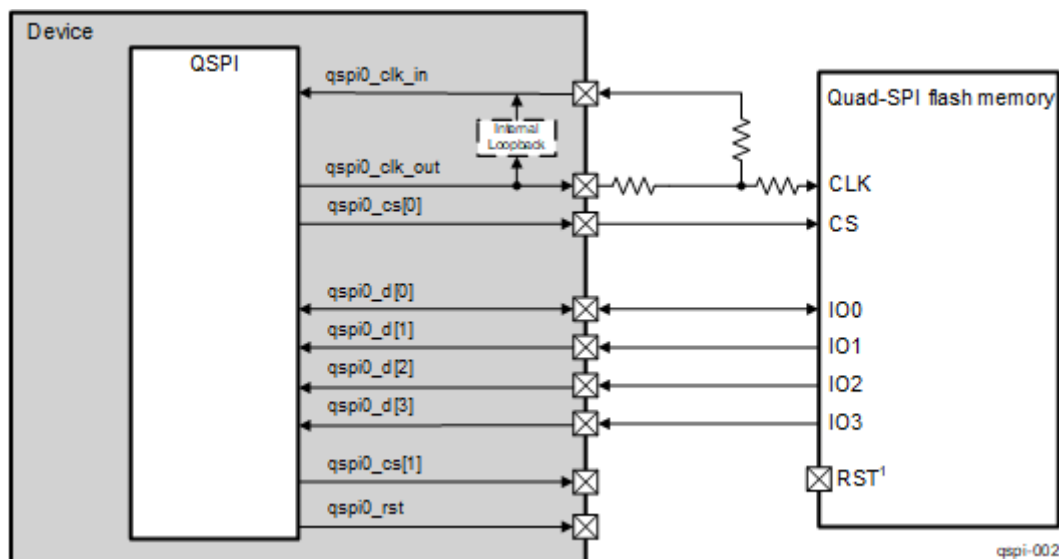


Figure 10-2. QSPI Connected to an External Quad-SPI Flash Memory

1. External flash memories that are larger than 128 Mb require an external reset pin for correct operation after SoC PORz reset. This reset must be triggered upon board reset to ensure the flash is in the correct state upon boot.

Table 10-1 lists and describes the QSPI I/O signals.

Table 10-1. QSPI I/O Signals

QSPI Signal/Pad name	I/O <sup>(1)</sup>	Description					
		3-pin <sup>(2)</sup> SPI Read (Single Read)	3-pin <sup>(2)</sup> SPI Write (Single Write)	4-pin <sup>(2)</sup> SPI Read (Single Read)	4-pin <sup>(2)</sup> SPI Write (Single Write)	4-pin <sup>(2)</sup> SPI Read (Dual Read)	6-pin <sup>(2)</sup> SPI Read (Quad Read)
qspi0_d[0]	IO	Used as SPI data input	Used as SPI data output	Not used	Used as SPI data output	Used as SPI data input 0	Used as SPI data input 0
qspi0_d[1]	I	Not used	Not used	Used as SPI data input	Not used	Used as SPI data input 1	Used as SPI data input 1
qspi0_d[2]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 2
qspi0_d[3]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 3

**Table 10-1. QSPI I/O Signals (continued)**

QSPI Signal/Pad name	I/O <sup>(1)</sup>	Description					
		3-pin <sup>(2)</sup> SPI Read (Single Read)	3-pin <sup>(2)</sup> SPI Write (Single Write)	4-pin <sup>(2)</sup> SPI Read (Single Read)	4-pin <sup>(2)</sup> SPI Write (Single Write)	4-pin <sup>(2)</sup> SPI Read (Dual Read)	6-pin <sup>(2)</sup> SPI Read (Quad Read)
qspi0_sclk	O	Clock for the external SPI device					
qspi0_cs[0]	O	External SPI device chip-select 0					
qspi0_rtclock	I	The qspi0_sclk output must be connected to the qspi0_rtclock input, and is used for controlling the timing of the read return data when the QSPI module operates in Mode 0. In case Mode 3 is used, there is no need to connect the qspi0_sclk to the qspi0_rtclock.					

(1) I = Input; O = Output

(2) This is the pin count at the SPI flash memory side. The pin count at the device side is increased by one because of the qspi0\_rtclock signal. References to the pin count throughout this chapter consider the pin count at the SPI flash memory side.

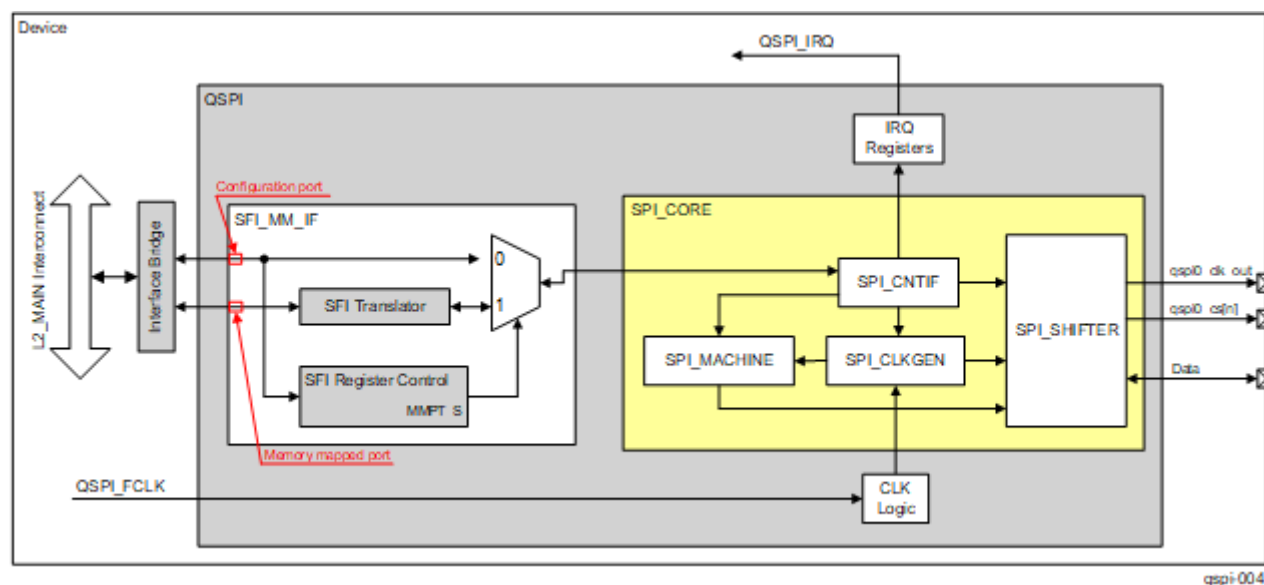
### Note

To ensure proper timing, precise layout and routing requirements must be followed. For layout and routing requirements for all QSPI signals, see section “PCB Guidelines” of the device Data Manual.

## 10.3 QSPI Functional Description

### 10.3.1 QSPI Block Diagram

Initial device boot from external SPI flash memory can be accomplished through the QSPI module. The interface is a simple 4-wire SPI used for control or data transfers. The QSPI also supports a 3-wire SPI protocol where the qspi0\_d[0] signal is used as a bidirectional for reads and writes. In addition, a 6-wire mode can be used to support quad read devices. Figure 10-3 shows the QSPI block diagram.


**Figure 10-3. QSPI Block Diagram**

The QSPI is composed of two blocks. The first one is the SFI memory-mapped interface (SFI\_MM\_IF) and the second one is the SPI core (SPI\_CORE). The SFI\_MM\_IF block is associated only with SPI flash memories and is used for specifying typical for the SPI flash memories settings (read or write command, number of address and dummy bytes, and so on) unlike the SPI\_CORE block, which is associated with the SPI interface itself and is used to configure typical SPI settings (chip-select polarity, serial clock inactive state, SPI clock mode, length of the words transferred, and so on).

The SFI\_MM\_IF comprises the following two subblocks:

- SFI register control
- SFI translator

The SPI\_CORE comprises the following four subblocks:

- SPI control interface (SPI\_CNTIF)
- SPI clock generator (SPI\_CLKGEN)
- SPI control state machine (SPI\_MACHINE)
- SPI data shifter (SPI\_SHIFTER)

In addition, an interface bridge connects the two ports (configuration port and memory-mapped port) of the SFI\_MM\_IF block to the L2\_MAIN interconnect. There are no software controls associated with this interface bridge.

The QSPI supports long transfers through a frame-style sequence. In its generic SPI use mode, a word can be defined up to 128 bits and multiple words can be transferred during a single access. For each word, a device initiator must read or write the new data and then tell the QSPI to continue the current operation. Using this sequence, a maximum of 4096 128-bit words can be transferred in a single SPI read or write operation. This allows great flexibility when connecting the QSPI to various types of devices.

As opposed to the generic SPI use mode, the communication with serial flash-type devices requires sending a byte command, followed by sending bytes of data. Commands can be sent through the SPI\_CORE block to communicate with a serial flash device; however, it is easier to do this using the SFI\_MM\_IF block because it is intended to ease the communication with serial flash devices. If the SPI\_CORE is used to communicate with a serial flash device, software must load the command into the SPI data transfer register with additional configuration fields, perform the byte transfer, then place the data to be sent (or configure for receive) along with additional configuration fields, and perform that transfer. Reads and writes to serial flash devices are more specific. First, the read or write command byte is sent, followed by 1 to 4 bytes of address (corresponding to the address to read/write), then followed by the data write/receive phase. Data is always sent byte oriented. When the address is loaded, data can be continuously read or written, and the address will automatically increment to each byte address internally to the serial flash device.

#### Note

The SFI\_MM\_IF block only allows reading and writing to an externally connected SPI flash device. The SFI\_MM\_IF block does not allow reads or writes to internal configuration and status registers of the SPI flash device. These registers must be accessed through the features of the SPI\_CORE block.

#### 10.3.1.1 SFI Register Control

The SFI register control block consists of the following five configuration registers:

- QSPI\_SPI\_SETUP0\_REG
- QSPI\_SPI\_SETUP1\_REG
- QSPI\_SPI\_SETUP2\_REG
- QSPI\_SPI\_SETUP3\_REG
- QSPI\_SPI\_SWITCH\_REG

The first four registers let the user define the following:

- Byte command for a serial flash read specified by the QSPI\_SPI\_SETUP<sub>i</sub>\_REG[7:0] RCMD bit field
- Byte command for a serial flash write specified by the QSPI\_SPI\_SETUP<sub>i</sub>\_REG[23:16] WCMD bit field
- Number of address bytes required for the particular type of serial flash specified by the QSPI\_SPI\_SETUP<sub>i</sub>\_REG[9:8] NUM\_A\_BYTES bit field
- Number of "dummy bytes" that may be needed to support the fast read mode function of some serial flash devices. The QSPI\_SPI\_SETUP<sub>i</sub>\_REG[11:10] NUM\_D\_BYTES bit field specifies the number of "dummy bits." In addition, the QSPI\_SPI\_SETUP<sub>i</sub>\_REG[28:24] NUM\_D\_BITS bit field can also specify the number of "dummy bits."

- Whether the read command is single (normal), dual, or quad read mode command. This is specified by the QSPI\_SPI\_SETUP<sub>i</sub>\_REG[13:12] READ\_TYPE bit field.
- *i* is equal to 0, 1, 2 and 3 and means that the QSPI\_SPI\_SETUP<sub>i</sub>\_REG registers are associated with each of the four supported chip-selects [that is, four supported output SPI flash devices])

The QSPI\_SPI\_SWITCH\_REG register acts as a static switch which allows the configuration port (shown in Figure 10-3) to connect directly to the SPI\_CORE block, or allows the memory-mapped port (also shown in Figure 10-3) to connect to the SPI\_CORE block. This is done using the QSPI\_SPI\_SWITCH\_REG[0] MMPT\_S bit.

In addition, the QSPI\_SPI\_SWITCH\_REG[1] MM\_INT\_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

### 10.3.1.2 SFI Translator

The SFI translator block represents an FSM which, based on the configuration information loaded into the SFI register control block, converts each input read/write sequence into an SPI\_CORE configuration sequence for access to the external serial flash memory.

A read sequence is converted into the following actions:

1. SPI chip-select goes active.
2. Read command byte is issued.
3. 1 to 4 address bytes, which correspond to the first address supplied, are issued.
4. 0 to 3 dummy bytes are issued, if “fast read” is supported.
5. Data bytes are read from the external SPI flash memory.
6. SPI chip-select goes inactive.

For linear addressing mode, action 5 is repeated until the byte count to be transferred reaches zero.

A write sequence is identical to a read sequence, except that a write sequence does not use dummy bytes.

Another important aspect with regard to writes is that a serial flash memory location can only be written to if the bits are erased in advance. Erased means the bits are set to 1. This means that writing only changes 1 contents to 0. It is not possible with this write to change the contents of a bit from 0 to 1. An erase command must be performed to do this operation. Erase commands cannot be executed on single byte locations. Depending on device types, there are page, block, and chip erase commands. To perform an erase command, the particular command must be sent over the SPI bus, and an internal register of the serial flash device must then be polled to determine when the erase completes. The erases must be done through the configuration port by software before performing any writes through the memory-mapped port. This means that writes are passed through to the serial flash device, but if the memory locations being modified are not properly erased before the write, the contents may not result in what was sent.

#### Note

The input to the SFI Memory Mapped Protocol Translator is 23 address lines. Therefore, the SFI mode of operation supports external flash size of up to 8MB

### 10.3.1.3 SPI Control Interface

The SPI control interface contains configuration registers used to configure the SPI core functionality of the QSPI. This block maintains all configuration settings for the SPI core (that is, settings specific for the SPI interface itself but not for the SPI flash memories).

The registers defined for this block are:

- The QSPI\_PID register, which is read only and contains QSPI revision associated information
- The QSPI\_SPI\_CLOCK\_CNTRL\_REG register, which is used to control external SPI clock (qspi0\_sclk)
- The QSPI\_SPI\_DC\_REG register used to define the SPI clock mode and chip-select polarity for the four external SPI devices



- The QSPI\_SPI\_CMD\_REG register used to control the operation of the SPI command. This register is also used to configure and transfer data.
  - Four data registers used for reading the data received and for writing the data to be transferred. These registers are:
    - QSPI\_SPI\_DATA\_REG
    - QSPI\_SPI\_DATA\_REG\_1
    - QSPI\_SPI\_DATA\_REG\_2
    - QSPI\_SPI\_DATA\_REG\_3
- These four registers compose a 128-bit shift register.
- The QSPI\_SPI\_STATUS\_REG register, which contains status information

All of these registers can only be written if the QSPI is not busy. This means that they can be written if the QSPI\_SPI\_STATUS\_REG[0] BUSY bit is 0x0. The QSPI becomes busy when a write to the QSPI\_SPI\_CMD\_REG[18:16] CMD bit field is performed. Writing to this bit field starts an SPI transaction and sets the QSPI\_SPI\_STATUS\_REG[0] BUSY bit to 0x1. The CMD bit field can be written again when the BUSY bit is 0x0. In addition, the start of the SPI transaction is synchronized to the qspi0\_sclk clock and clearing of the BUSY bit is synchronized to the QSPI\_FCLK clock.

The register group QSPI\_SPI\_DATA\_REG\_3, QSPI\_SPI\_DATA\_REG\_2, QSPI\_SPI\_DATA\_REG\_1 and QSPI\_SPI\_DATA\_REG is treated as a single 128-bit word for shifting data in and out. The QSPI\_SPI\_DATA\_REG\_3 register is used for the most significant bits and the QSPI\_SPI\_DATA\_REG is used for the least significant bits. This applies for both reads and writes. For example, after reading a 128-bit word (WLEN = 0x7F) the most significant bit of the data read, that is bit 127, will be located at QSPI\_SPI\_DATA\_REG\_3[31] position and the least significant bit, that is bit 0 of the data read, will be located at the QSPI\_SPI\_DATA\_REG[0] position.

The data written to this register group should be right justified so that a data pre-shifting is not required. The QSPI\_SPI\_CMD\_REG[25:19] WLEN bit field determines the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the WLEN bit field should be set to 0x7 and the data byte should be written to the lower byte of the QSPI\_SPI\_DATA\_REG register. By setting the word length to 0x7 the QSPI\_SPI\_DATA\_REG register will look like a pseudo 8-bit shift register. When the user wants to write 40-bit long word the WLEN bit field should be set to 0x27, the 32 least significant bits of data should be written to the QSPI\_SPI\_DATA\_REG and the rest 8 most significant bits of data should be written to the lower byte of the QSPI\_SPI\_DATA\_REG\_1 register. By setting WLEN to 0x27 these two registers will look like a pseudo 40-bit shift register. When the word length is greater than 64 bits the QSPI\_SPI\_DATA\_REG\_2 register is also used and the previously described logic applies. The QSPI\_SPI\_DATA\_REG\_3 register is used together with the other three data registers when the word length is greater than 96 bits.

When dual or quad read mode is used the number of the words transferred must be even. This number is configured through the QSPI\_SPI\_CMD\_REG[11:0] FLEN bit field.

#### Note

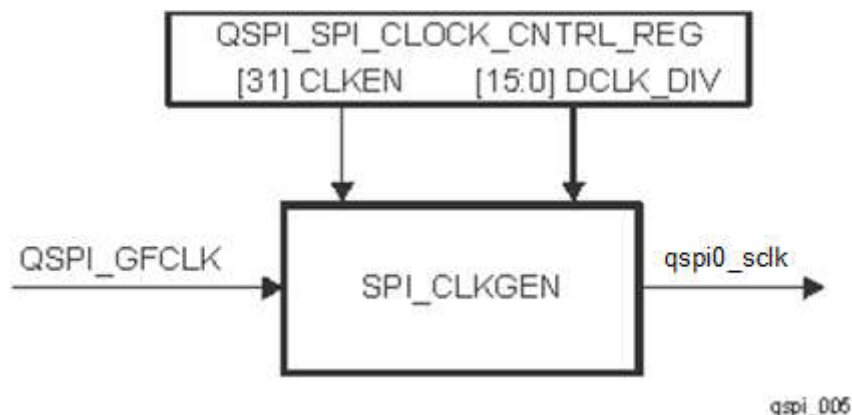
The QSPI module does not support a "pass through" mode where the data present on qspi0\_d[1] is sent to qspi0\_d[0], when 4-pin non-dual read mode is used. This means that setting the QSPI\_SPI\_CMD\_REG[18:16] CMD bit field to 0x1 causes the QSPI only to read from an external device using the qspi0\_d[1] pad as an input and if a write to the same external device is desired, the CMD bit field should be set to 0x2, which causes the qspi0\_d[0] pad to be used as an output.

#### 10.3.1.4 SPI Clock Generator

The SPI clock generator uses the QSPI\_FCLK clock as an input, and generates the qspi0\_sclk, which is a divided version of the QSPI\_FCLK clock. The divide ratio is a 16-bit value configured through the QSPI\_SPI\_CLOCK\_CNTRL\_REG[15:0] DCLK\_DIV bit field and thus provides a division factor in a range from 1 to 65536. The QSPI\_FCLK clock is divided by the DCLK\_DIV value + 1 to provide the qspi0\_sclk clock. When DCLK\_DIV = 0x0 the QSPI\_FCLK clock equals the DCLK clock. The value in the DCLK\_DIV bit field

applies only when the QSPI\_SPI\_CLOCK\_CNTRL\_REG[31] CLKEN bit is set to 0x1. Figure 10-4 shows the SPI\_CLKGEN block.

If the CLKEN bit is 0x0 the command specified in the QSPI\_SPI\_CMD\_REG[18:16] CMD bit field is not executed and the QSPI\_SPI\_STATUS\_REG[0] BUSY bit is not set. The command is executed only if the CLKEN bit is 0x1 before write to the CMD bit field.



**Figure 10-4. SPI\_CLKGEN Block**

### 10.3.1.5 SPI Control State-Machine

The SPI control state-machine (SPI\_MACHINE) manages the operation of the SPI\_CORE block. SPI\_MACHINE takes control and configuration information from the registers in the SPI\_CNTIF block as input and provides control information to the SPI data shifter. This information is used to control the SPI data port. The SPI\_MACHINE also generates status information, which is sent back to the SPI\_CNTIF block.

Writing a valid value to the QSPI\_SPI\_CMD\_REG[18:16] CMD bit field sets immediately the QSPI\_SPI\_STATUS\_REG[0] BUSY bit to 0x1, activates the corresponding qspi0\_cs[n] (n = 0 to 1) and starts the SPI data transaction. The BUSY bit is cleared automatically when QSPI\_SPI\_CMD\_REG[25:19] WLEN number of bits are shifted in or out. If the value of the QSPI\_SPI\_STATUS\_REG[27:16] WDCNT bit field is different than 0x0 and WLEN number of bits are shifted already, the SPI\_MACHINE waits until another write to the CMD bit field is performed. If the command written to the CMD bit field is valid, then this increments the value of the WDCNT bit field from 0x0 and starts shifting data in or out again. This is repeated until the WDCNT bit field reaches the frame length (QSPI\_SPI\_CMD\_REG[11:0] FLEN), that is, all words of the frame are shifted or till earlier frame termination occurs. While the SPI\_MACHINE is waiting for write to the CMD bit field the corresponding qspi0\_cs[n] (n = 0 to 1) remains active and the BUSY flag is set to 0x0. In addition, the bit length for each word can be changed during a frame from 1 to 128 bits using the QSPI\_SPI\_CMD\_REG[25:19] WLEN bit field.

The SPI\_MACHINE also provides a mechanism to terminate the frame earlier. This is done by writing an invalid command to the CMD bit field. An invalid command corresponds to the 0x0 and 0x4 (reserved) values of the CMD bit field. Writing one of these values when the the WDCNT bit field is not equal to 0x0 and when the BUSY flag is 0x0 terminates the frame earlier.

The corresponding qspi0\_cs[n] (n = 0 to 1) becomes inactive when all words are shifted or when the frame terminates earlier.

### 10.3.1.6 SPI Data Shifter

The SPI data shifter handles the capture and generation of the SPI interface signals. Based on control signals from the SPI\_MACHINE and SPI\_CNTIF blocks, data is shifted in or out on falling or rising edge of qspi0\_sclk clock depending on the SPI clock mode selected. Table 10-2 lists the four defined clock modes of operation for the QSPI.

**Table 10-2. SPI Clock Modes Definition**

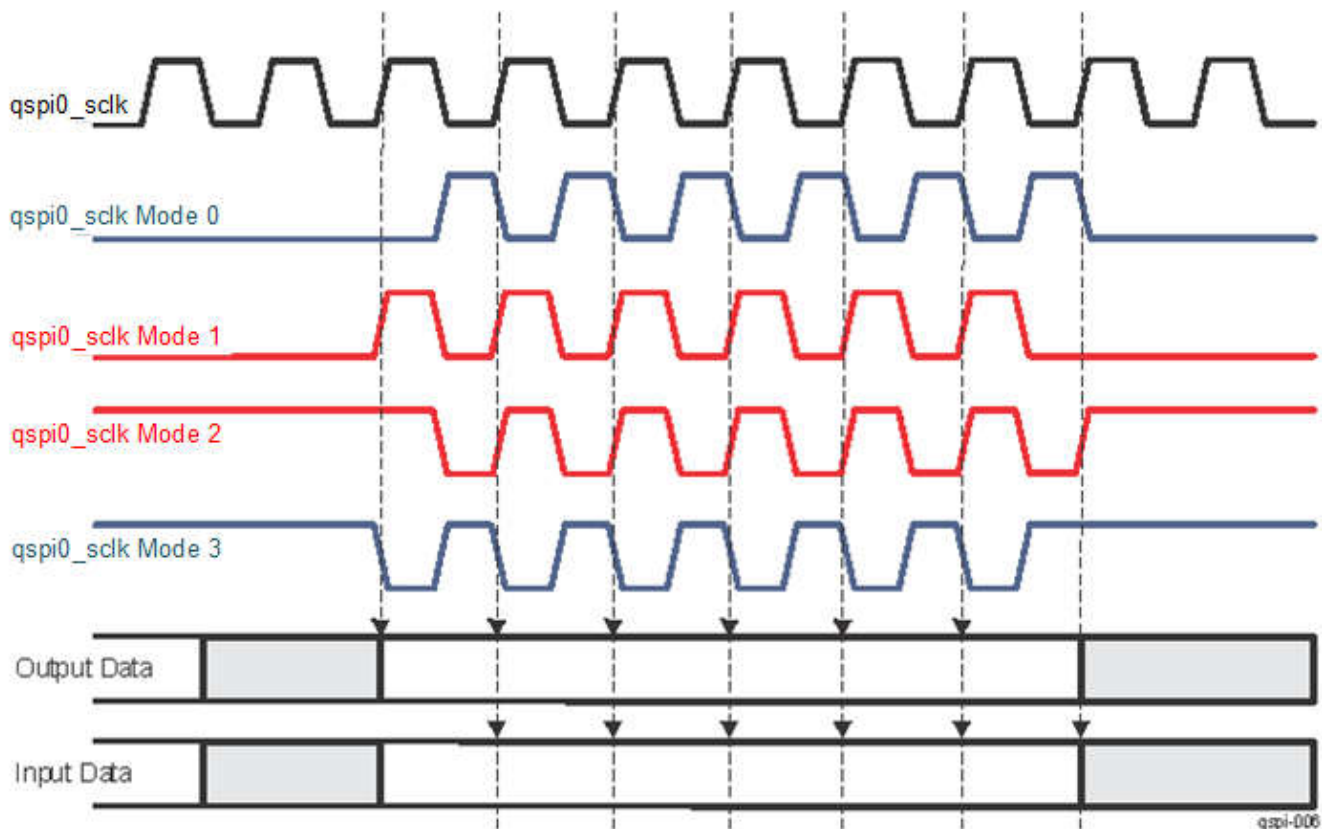
Mode	Settings in the QSPI_SPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
0	0	0	Data input captured on falling edge of qspi0_sclk clock. Data output generated on falling edge of qspi0_sclk clock
1	0	1	Data input captured on rising edge of qspi0_sclk clock. Data output generated on rising edge of qspi0_sclk clock
2	1	0	Data input captured on rising edge of qspi0_sclk clock. Data output generated on rising edge of qspi0_sclk clock
3	1	1	Data input captured on falling edge of qspi0_sclk clock. Data output generated on falling edge of qspi0_sclk clock

**Note**

Mode 1 and Mode 2 are not supported and should not be used.

The CKPi and CKPHi (i = 0 to 3) bits of the QSPI\_SPI\_DC\_REG register control the clock modes. Each of these 4 bits corresponds to an output chip select.

Figure 10-5 shows all four clock modes. In addition, through the DDi (i = 0 to 3) bits of the QSPI\_SPI\_DC\_REG register the data can be delayed from one to three qspi0\_sclk clock cycles after the corresponding qspi0\_cs[n] (n = 0 to 1) goes active. The active state of each chip-select can also be controlled through the CSPi (i = 0 to 3) bits of the QSPI\_SPI\_DC\_REG register.



**Figure 10-5. SPI Clock Modes**

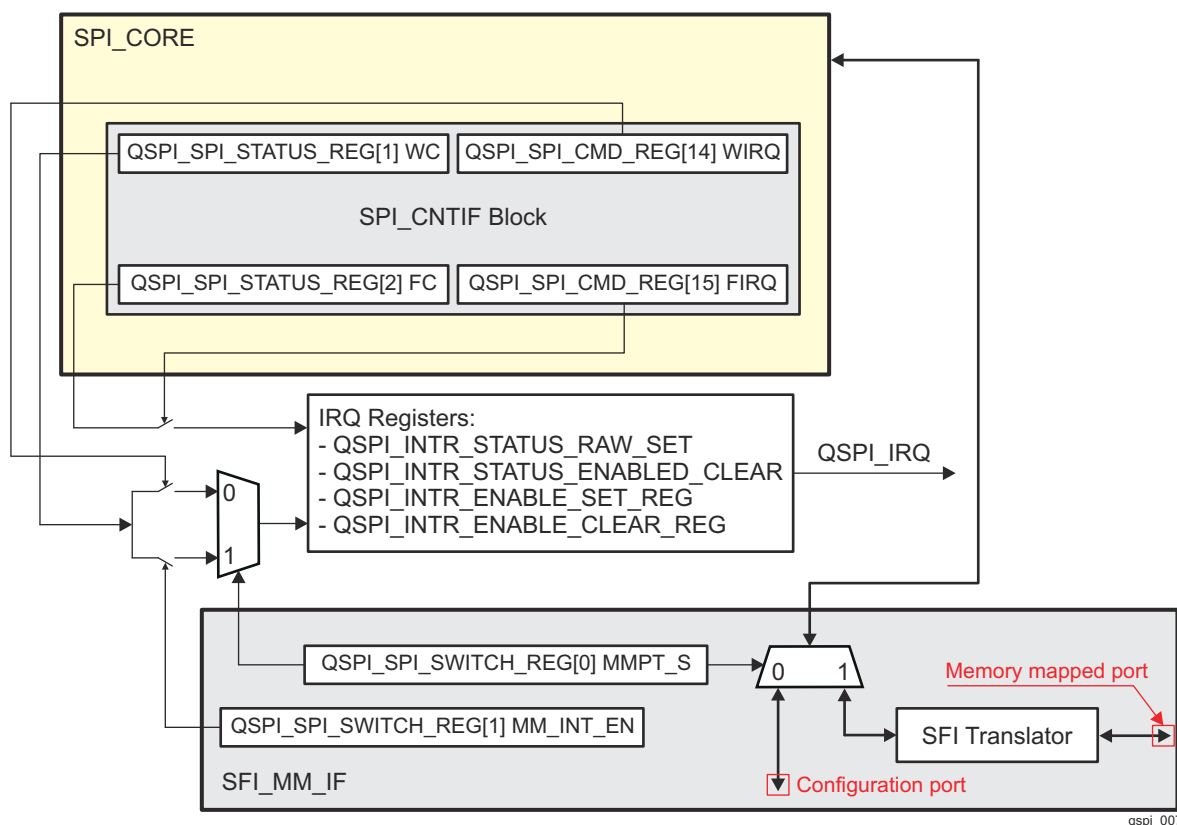
### 10.3.2 QSPI Clock Configuration

The QSPI complies with the PRCM peripheral-idle protocol. The QSPI\_FCLK clock is gated based on the values loaded in the QSPI\_SYSCONFIG[3:2] IDLE\_MODE bit field. Three modes are supported:

- Force-idle: The QSPI\_FCLK clock is gated unconditionally by the QSPI.
- No-idle: The QSPI\_FCLK clock is never gated by the QSPI.
- Smart-idle: The QSPI\_FCLK clock is gated by the QSPI, depending on its internal requirements.

### 10.3.3 QSPI Interrupt Requests

Figure 10-6 shows a logical representation of the QSPI interrupt generation scheme.



**Figure 10-6. Logical Representation of the QSPI Interrupt Generation Scheme**

QSPI\_SPI\_STATUS\_REG[1] WC and QSPI\_SPI\_STATUS\_REG[2] FC are status bits indicating whether word or frame transfer is complete. Setting the corresponding interrupt enable bit (WIRQ or FIRQ) in the QSPI\_SPI\_CMD\_REG register allows these events (WC and FC) to generate an interrupt. The WC and FC bits are reset every time the user writes to the QSPI\_SPI\_CMD\_REG register or reads the QSPI\_SPI\_STATUS\_REG register. This is done to keep control parameters from changing the interface protocol signals while a transfer is in progress. Additionally, the QSPI\_SPI\_SWITCH\_REG[1] MM\_INT\_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

When the QSPI\_SPI\_CMD\_REG[14] WIRQ and QSPI\_SPI\_CMD\_REG[15] FIRQ bits are set to 0x1 the following applies:

- The QSPI activates its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the QSPI\_INTR\_ENABLE\_SET\_REG register. These interrupts can be disabled by setting the corresponding bits in the QSPI\_INTR\_ENABLE\_CLEAR\_REG register to 0x1.
- After an interrupt has been serviced, software must clear the corresponding status flag. This is done by setting the corresponding bit in the QSPI\_INTR\_STATUS\_ENABLED\_CLEAR register to 0x1, which

also clears the corresponding bit in the QSPI\_INTR\_STATUS\_RAW\_SET register. The status flags in the QSPI\_INTR\_STATUS\_RAW\_SET register are set even if the corresponding interrupt is disabled unlike those in the QSPI\_INTR\_STATUS\_ENABLED\_CLEAR register, which are set only if the corresponding interrupt is enabled.

- The QSPI also generates an interrupt if a certain bit in the QSPI\_INTR\_STATUS\_RAW\_SET register is set to 0x1 and the corresponding interrupt is enabled through the QSPI\_INTR\_ENABLE\_SET\_REG register. This feature is useful during user software debugging. In addition, even if interrupts are not enabled a corresponding raw flag in the QSPI\_INTR\_STATUS\_RAW\_SET register is set to 0x1 when an IRQ condition occurs.
- Even if interrupts are not enabled, a certain status bit in the QSPI\_INTR\_STATUS\_RAW\_SET register can also be cleared by setting to 0x1 the corresponding bit in the QSPI\_INTR\_STATUS\_ENABLED\_CLEAR register.

It must be considered that the previously described scenario applies if the QSPI\_SPI\_CMD\_REG[14] WIRQ and QSPI\_SPI\_CMD\_REG[15] FIRQ bits are set to 0x1.

#### Note

The QSPI\_IRQ interrupt line is activated only if at least one of the following conditions is met:

- The word complete interrupt is enabled:
  - during operations using the memory-mapped port by setting to 0x1 both the QSPI\_SPI\_SWITCH\_REG[1] MM\_INT\_EN and QSPI\_INTR\_ENABLE\_SET\_REG[1] WIRQ\_ENA\_SET bits.
  - during operations using the configuration port by setting to 0x1 both the QSPI\_SPI\_CMD\_REG[14] WIRQ and QSPI\_INTR\_ENABLE\_SET\_REG[1] WIRQ\_ENA\_SET bits.
- The frame complete interrupt is enabled setting to 0x1 both the QSPI\_SPI\_CMD\_REG[15] FIRQ and QSPI\_INTR\_ENABLE\_SET\_REG[0] FIRQ\_ENA\_SET bits.

The QSPI\_IRQ interrupt line is also activated when both the conditions are met.

Table 10-3 lists the event flags and the corresponding mask bits of the sources which can cause interrupts.

**Table 10-3. QSPI Events**

Event Flag	Event Mask	Description
QSPI_INTR_STATUS_RAW_SET[1] WIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[1] WIRQ_ENA QSPI_SPI_STATUS_REG[1] WC	QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[1] WIRQ_ENA_CLR QSPI_SPI_CMD_REG[14] WIRQ	Word complete interrupt event. Asserted each time after a word is transferred or received.
QSPI_INTR_STATUS_RAW_SET[0] FIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[0] FIRQ_ENA QSPI_SPI_STATUS_REG[2] FC	QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[0] FIRQ_ENA_CLR QSPI_SPI_CMD_REG[15] FIRQ	Frame complete interrupt event. Asserted each time after a frame is transferred or received.

#### Note

QSPI\_IRQ can also be used to trigger DMA events

### 10.3.4 QSPI Memory Regions

Two memory regions are associated with the QSPI. The first memory region is dedicated to the configuration port. Using this memory region, all internal registers can be programmed and serial transfers made from the supported external SPI devices. The L2\_MAIN start address at which the configuration port is available is

0x4820 0000. The second memory region is associated mainly with the memory-mapped port and is used for communication directly with one of the two supported external SPI devices. The memory region for device 1 starts at 0x6000 0000 and the memory region for device 2 starts at 0x6200 0000

It is important to keep in mind that the configuration port provides an access to all the QSPI registers listed in the register summary. These are configuration registers and also four data registers. The configuration registers are used to configure typical SPI and serial flash memory settings and the four data registers are used for read and write operations. When communicating with an external SPI device (but not an SPI flash memory) the SPI\_CORE module should be used and the data exchanged is available through these four data registers, which can be accessed only through the configuration port.

## 10.4 QSPI Registers

### 10.4.1 QSPI Register Summary

**Table 10-4. QSPI Registers Mapping Summary**

Register Name	Type	Register Width (Bits)	Address Offset
QSPI_PID	R	32	0x0000 0000
QSPI_SYSCONFIG	RW	32	0x0000 0010
QSPI_INTR_STATUS_RAW_SET	RW	32	0x0000 0020
QSPI_INTR_STATUS_ENABLED_CLEAR	RW	32	0x0000 0024
QSPI_INTR_ENABLE_SET_REG	RW	32	0x0000 0028
QSPI_INTR_ENABLE_CLEAR_REG	RW	32	0x0000 002C
QSPI_INTC_EOI_REG	RW	32	0x0000 0030
QSPI_SPI_CLOCK_CNTRL_REG	RW	32	0x0000 0040
QSPI_SPI_DC_REG	RW	32	0x0000 0044
QSPI_SPI_CMD_REG	RW	32	0x0000 0048
QSPI_SPI_STATUS_REG	R	32	0x0000 004C
QSPI_SPI_DATA_REG	RW	32	0x0000 0050
QSPI_SPI_SETUP0_REG	RW	32	0x0000 0054
QSPI_SPI_SETUP1_REG	RW	32	0x0000 0058
QSPI_SPI_SETUP2_REG	RW	32	0x0000 005C
QSPI_SPI_SETUP3_REG	RW	32	0x0000 0060
QSPI_SPI_SWITCH_REG	RW	32	0x0000 0064
QSPI_SPI_DATA_REG_1	RW	32	0x0000 0068
QSPI_SPI_DATA_REG_2	RW	32	0x0000 006C
QSPI_SPI_DATA_REG_3	RW	32	0x0000 0070

### 10.4.2 QSPI Register Description

**Table 10-5. QSPI\_PID**

<b>Address Offset</b>	0x0000 0000
<b>Description</b>	Revision register
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal data

**Table 10-6. QSPI\_SYSCONFIG**

<b>Address Offset</b>	0x0000 0010
<b>Description</b>	
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLE_	RESE		
																												MODE	RVED		

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x2



Bits	Field Name	Description	Type	Reset
3:2	IDLE_MODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Reserved.	RW	0x2
1:0	RESERVED		R	0x0

**Table 10-7. QSPI\_INTR\_STATUS\_RAW\_SET**
**Address Offset** 0x0000 0020

**Description** This register contains raw interrupt status flags.

**Type** RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WI	FI			
																											R	R			
																											Q_	Q_			
																											RA	RA			
																											W	W			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0
1	WIRQ_RAW	Word Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0
0	FIRQ_RAW	Frame Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0

**Table 10-8. QSPI\_INTR\_STATUS\_ENABLED\_CLEAR**
**Address Offset** 0x0000 0024

**Description** This register contains status flags of the enabled interrupts.

**Type** RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WI R Q EN A	FI R Q EN A			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0



Bits	Field Name	Description	Type	Reset
1	WIRQ_ENA	Word Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the word interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0
0	FIRQ_ENA	Frame Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the frame interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0

**Table 10-9. QSPI\_INTR\_ENABLE\_SET\_REG**

<b>Address Offset</b>	0x0000 0028
<b>Description</b>	This register enables the interrupts.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0		
RESERVED																																WIRQ_ENA_SET	FIRQ_ENA_SET

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_SET	Word interrupt enable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt enabled Write: 0x0: Has no effect 0x1: Enables the word interrupt	RW	0x0
0	FIRQ_ENA_SET	Frame interrupt enable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Enables the frame interrupt	RW	0x0

**Table 10-10. QSPI\_INTR\_ENABLE\_CLEAR\_REG**

<b>Address Offset</b>	0x0000 002C
<b>Description</b>	This register disables the interrupts.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WIRQ_ENA_CLR		FIRQ_ENA_CLR													

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_CLR	Word interrupt disable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the word interrupt	RW	0x0
0	FIRQ_ENA_CLR	Frame interrupt disable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the frame interrupt	RW	0x0

**Table 10-11. QSPI\_INTC\_EOI\_REG**

<b>Address Offset</b>	0x0000 0030
<b>Description</b>	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if a new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the interrupt outputs. There is one interrupt output. Write 0x0 after servicing the interrupt to be able to generate another interrupt if pulse interrupts are used. Any other write value is ignored.	RW	0x0

**Table 10-12. QSPI\_SPI\_CLOCK\_CNTRL\_REG**

<b>Address Offset</b>	0x0000 0040
<b>Description</b>	This register controls the external SPI clock generation. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.

**Table 10-12. QSPI\_SPI\_CLOCK\_CNTRL\_REG (continued)**

RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CL KE N	RESERVED															DCLK_DIV															

Bits	Field Name	Description	Type	Reset
31	CLKEN	External SPI clock (qspi1_sclk) enable. 0x0: The qspi1_sclk clock is turned off 0x1: The qspi1_sclk clock is enabled	RW	0x0
30:16	RESERVED		R	0x0
15:0	DCLK_DIV	Divide ratio for the external SPI clock (qspi1_sclk)	RW	0x0

**Table 10-13. QSPI\_SPI\_DC\_REG**
**Address Offset** 0x0000 0044

**Description** This register controls the different modes for each output chip select. This register can only be written when the QSPI module is not busy, as identified by the QSPI\_SPI\_STATUS\_REG[0] BUSY bit.

**Type** RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DD0				CK PH 0	CS P0	CK P0	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:3	DD0	Data delay for chip select 0 0x0: Data is output on the same cycle as the qspi1_cs[0] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[0] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[0] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[0] goes active	RW	0x0
2	CKPH0	Clock phase for chip select 0. If CKP0 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP0 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
1	CSP0	Chip select polarity for chip select 0. 0x0: Active low 0x1: Active high	RW	0x0
0	CKP0	Clock polarity for chip select 0. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0

**Table 10-14. QSPI\_SPI\_CMD\_REG**

<b>Address Offset</b>	0x0000 0048
<b>Description</b>	This register sets up the SPI command. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESE RVED		CSNU M		RESE RVED													FI R Q	WI R Q	RESE RVED												

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	CSNUM	Device select. Sets the active chip select for the current transfer. 0x0: Chip Select 0 active 0x1: Chip Select 1 active 0x2: Chip Select 2 active 0x3: Chip Select 3 active	RW	0x0
27:26	RESERVED		R	0x0
25:19	WLEN	Word length. Sets the size of the individual transfers from 1 to 128 bits. When a word length greater than 32 bits is configured, not only the QSPI_SPI_DATA_REG register, but also the QSPI_SPI_DATA_REG_1, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_3 are used. One or all of these registers are used depending on the length of words transferred. 0x0: 1 bit 0x1: 2 bits ... 0x7F: 128 bits	RW	0x0
18:16	CMD	Transfer command. 0x0: Reserved 0x1: 4-pin Read Single 0x2: 4-pin Write Single 0x3: 4-pin Read Dual 0x4: Reserved 0x5: 3-pin Read Single 0x6: 3-pin Write Single 0x7: 6-pin Read Quad	RW	0x0
15	FIRQ	Frame complete interrupt enable. 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
14	WIRQ	Word complete interrupt enable 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
13:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:0	FLEN	Frame Length. 0x0: 1 word 0x1: 2 words ... 0xFFF: 4096 words	RW	0x0

**Table 10-15. QSPI\_SPI\_STATUS\_REG**

<b>Address Offset</b>	0x0000 004C
<b>Description</b>	This register contains indicators to allow the user to monitor the progression of a frame transfer. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
<b>Type</b>	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WDCNT												RESERVED												FC	W C	BU SY	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WDCNT	Word count. This field will reflect the 1-4096 words transferred	R	0x0
15:3	RESERVED		R	0x0
2	FC	Frame complete. This bit is set after the transmission of all the requested words completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Transfer is not complete 0x1: Transfer is complete	R	0x0
1	WC	Word complete. This bit is set after each word transfer completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Word transfer is not complete 0x1: Word transfer is complete	R	0x0
0	BUSY	Busy bit. Active transfer in progress. This bit is only set during an active word transfer. Between words it is cleared. 0x0: Idle 0x1: Busy	R	0x0

**Table 10-16. QSPI\_SPI\_DATA\_REG**

<b>Address Offset</b>	0x0000 0050
<b>Description</b>	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the first 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

**Table 10-17. QSPI\_SPI\_SETUP0\_REG**

<b>Address Offset</b>	0x0000 0054
<b>Description</b>	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 0 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVE D				NUM_D_BITS				WCMD								RESE RVED	READ_ TYPE	NUM_ D_BYT ES	NUM_ A_BYT ES	RCMD											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

**Table 10-18. QSPI\_SPI\_SETUP1\_REG**

<b>Address Offset</b>	0x0000 0058
<b>Description</b>	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 1 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVE D			NUM_D_BITS					WCMD								RESE RVED	READ_ TYPE	NUM_ D_BYT ES	NUM_ A_BYT ES	RCMD											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0

Bits	Field Name	Description	Type	Reset
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

**Table 10-19. QSPI\_SPI\_SETUP2\_REG**

<b>Address Offset</b>	0x0000 005C
<b>Description</b>	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 2 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				NUM_D_BITS				WCMD				RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0

Bits	Field Name	Description	Type	Reset
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

**Table 10-20. QSPI\_SPI\_SETUP3\_REG**

<b>Address Offset</b>	0x0000 0060
<b>Description</b>	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 3 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVE D			NUM_D_BITS					WCMD								RESE RVED	READ_ TYPE		NUM_ D_BYT ES		NUM_ A_BYT ES		RCMD								

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3



**Table 10-21. QSPI\_SPI\_SWITCH\_REG**

<b>Address Offset</b>	0x0000 0064
<b>Description</b>	This register allows initiators to switch control of the SPI core port between the configuration port and the SFI translator. In addition, an interrupt enable field is defined which is used to enable or disable word complete interrupt generation in memory mapped mode.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											M M_ I N T_ E N	M M_ P T_ S			

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	MM_INT_EN	Memory mapped mode interrupt enable.  0x0: Word complete interrupt is disabled during memory mapped operations 0x1: Word complete interrupt is enabled for memory mapped operations	RW	0x0
0	MMPT_S	MPT select.  0x0: Configuration port is selected to control the SPI_CORE. 0x1: SFI translator is selected to control the SPI_CORE.	RW	0x0

**Table 10-22. QSPI\_SPI\_DATA\_REG\_1**

<b>Address Offset</b>	0x0000 0068
<b>Description</b>	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the second 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

**Table 10-23. QSPI\_SPI\_DATA\_REG\_2**

<b>Address Offset</b>	0x0000 006C
<b>Description</b>	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the third 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.
<b>Type</b>	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

**Table 10-24. QSPI\_SPI\_DATA\_REG\_3**

Address Offset	0x0000 0070																																																																																																
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the fourth 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.																																																																																																
Type	RW																																																																																																
<table><tr><td>31</td><td>30</td><td>29</td><td>28</td><td>27</td><td>26</td><td>25</td><td>24</td><td>23</td><td>22</td><td>21</td><td>20</td><td>19</td><td>18</td><td>17</td><td>16</td><td>15</td><td>14</td><td>13</td><td>12</td><td>11</td><td>10</td><td>9</td><td>8</td><td>7</td><td>6</td><td>5</td><td>4</td><td>3</td><td>2</td><td>1</td><td>0</td></tr><tr><td colspan="32">DATA</td></tr></table>																																		31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	DATA																															
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0																																																																		
DATA																																																																																																	
Bits	Field Name	Description																				Type	Reset																																																																										
31:0	DATA	Data register for read and write operations																				RW	0x0																																																																										

## Chapter 11

# Enhanced Direct Memory Access (EDMA)

---



This section describes the Enhanced Direct Memory Access (EDMA) controller. For features applicable to the EDMA instances in the device, see the device-specific Integration section. The primary purpose of the EDMA controller is to service data transfers programmed between two memory-mapped follower endpoints on the device. The EDMA controller consists of two principle blocks:

- EDMA channel controllers: EDMA\_TPCC
- EDMA transfer controllers: EDMA\_TPTC

Devices can have multiple instances of EDMA channel controllers, each associated with multiple EDMA transfer controllers.

The EDMA channel controller serves as the user interface for the EDMA controller. The EDMA\_TPCC includes parameter RAM (PaRAM), channel control registers, and interrupt control registers. The EDMA\_TPCC serves to prioritize incoming software requests or events from peripherals, and submits transfer requests (TR) to the EDMA transfer controller.

The EDMA transfer controllers are responsible for data movement. The transfer request packets (TRP) submitted by the EDMA\_TPCC contain the transfer context, based on which the transfer controller issues read/write commands to the source and destination addresses programmed for a given transfer.

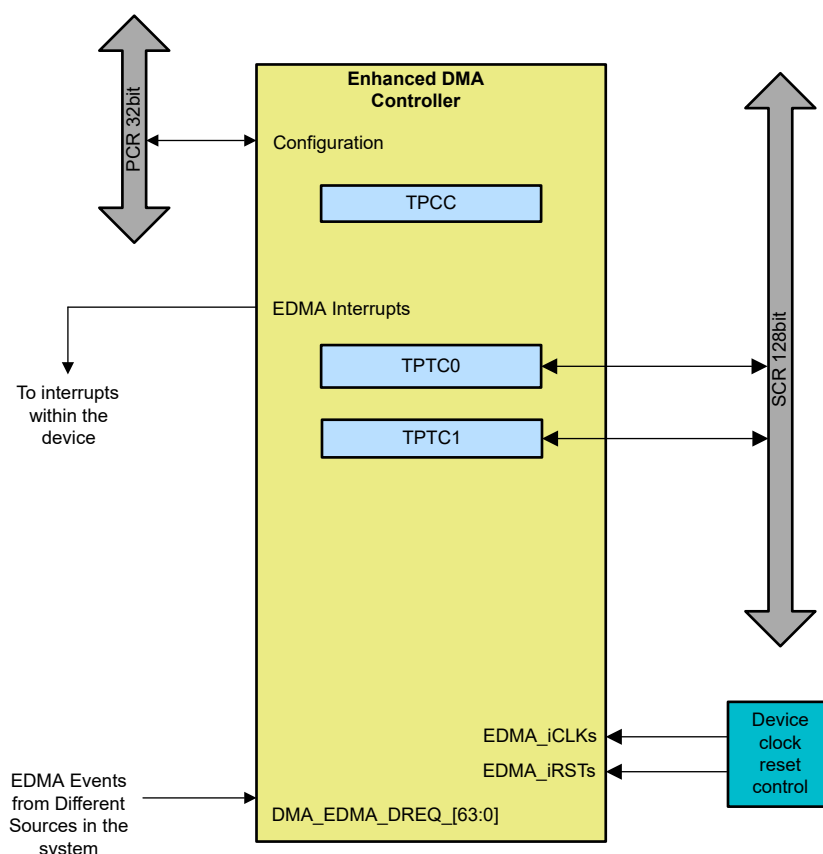
### 11.1 EDMA Module Overview

The enhanced direct memory access module, also called EDMA, performs high-performance data transfers between two target endpoints, memories and peripheral devices without microprocessor unit (MPU) or digital signal processor (DSP) support during transfer. EDMA transfer is programmed through a logical EDMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The EDMA controller is based on two major principal blocks:

- EDMA third-party channel controller (EDMA\_TPCC)
- EDMA third-party transfer controller (EDMA\_TPTC)

[Figure 11-1](#) shows an overview of the EDMA module.



**Figure 11-1. EDMA Module Overview**

For EDMA instances available on the device, see the device-specific integration section.

The **TPCC** is a high flexible channel controller that serves as both a user interface and an event interface for the EDMA controller. The EDMA\_TPCC serves to prioritize incoming software requests or events from peripherals, and submits transfer requests (TRs) to the transfer controller.

The **TPTC** performs read and write transfers by EDMA ports to the target peripherals, as programmed in the Active and Pending set of the registers. The transfer controllers are responsible for data movement, and issue read/write commands to the source and destination addresses programmed for a given transfer in the EDMA\_TPCC.

### 11.1.1 EDMA Features

This section shows generic EDMA features. For features applicable to the EDMA instances in the device, see the device-specific Integration section.

The EDMA\_TPCC channel controller has the following features:

- Fully orthogonal transfer description:
  - Three transfer dimensions
  - A-synchronized transfers: one dimension serviced per event
  - AB-synchronized transfers: two dimensions serviced per event
  - Independent indexes on source and destination
  - Chaining feature allowing a 3-D transfer based on a single event.
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes

- Linking mechanism allows automatic PaRAM set update
- Chaining allows multiple transfers to execute with one event
- Interrupt generation for the following:
  - Transfer completion
  - Error conditions
- Debug visibility:
  - Queue water marking/threshold
  - Error and status recording to facilitate debug
- 64 DMA request channels:
  - Event synchronization
  - Manual synchronization (CPUs write to event set registers EDMA\_TPCC\_ESR and EDMA\_TPCC\_ESRH).
  - Chain synchronization (completion of one transfer triggers another transfer).
- Eight QDMA channels:
  - QDMA channels trigger automatically upon writing to a parameter RAM (PaRAM) set entry.
  - Support for programmable QDMA channel to PaRAM mapping.
- Each PaRAM set can be used for a DMA channel, QDMA channel, or link set.
- Multiple transfer controllers/event queues.
- 16 event entries per event queue.

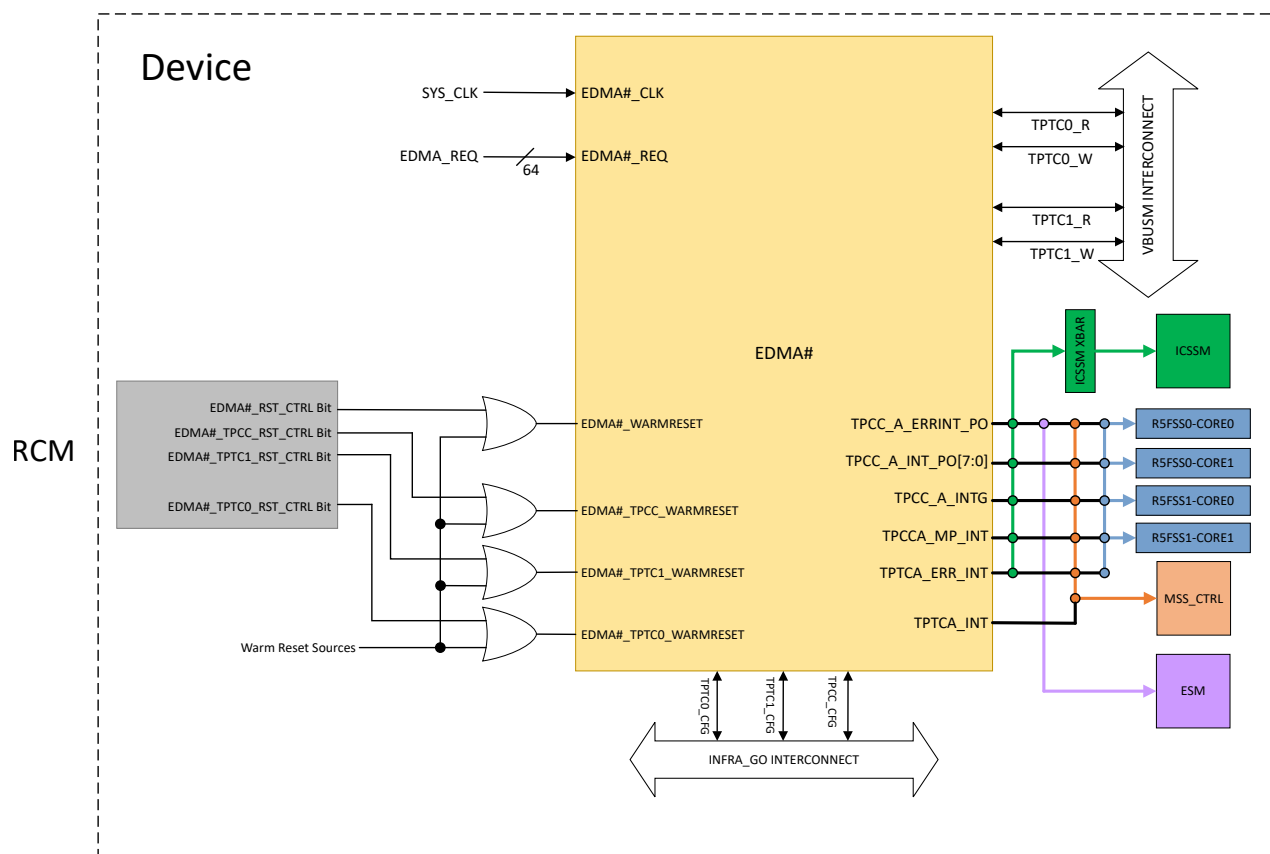
The **EDMA\_TPTC** transfer controller has the following features:

- 64-bit wide read and write ports per TC
- Supports two-dimensional transfers with independent indexes on source and destination (EDMA\_TPCC manages the third dimension)
- Support for increment or constant addressing mode transfers
- Interrupt and error support
- Memory-Mapped Register (MMR) bit fields are fixed position in 32-bit MMR regardless of endianness

## 11.2 EDMA Integration

This section describes modules integration in the device, including information about clocks, resets, and hardware requests.

## 11.2.1 EDMA Integration



**Figure 11-2. EDMA Integration Block Diagram**

### Note

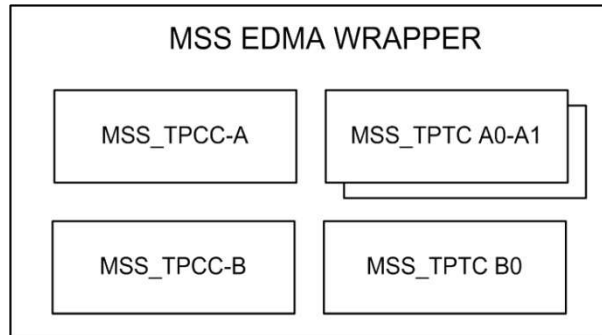
For more information on the interconnects, see the [System Interconnect](#) chapter.

For more information on power, reset, and clock management, see the corresponding sections within the [Device Configuration](#) chapter.

For more information on the device interrupt controllers, see the [Interrupt Controllers](#) chapter.

## 11.2.2 EDMA Integration in MSS

The MSS has 2 TPCCs and 3 TPTCs, as shown in [Figure 11-3](#).



**Figure 11-3. MSS EDMA Wrapper**

Table 11-1 and Table 11-2 list the configuration for the TPCCs and TPTCs in the MSS.

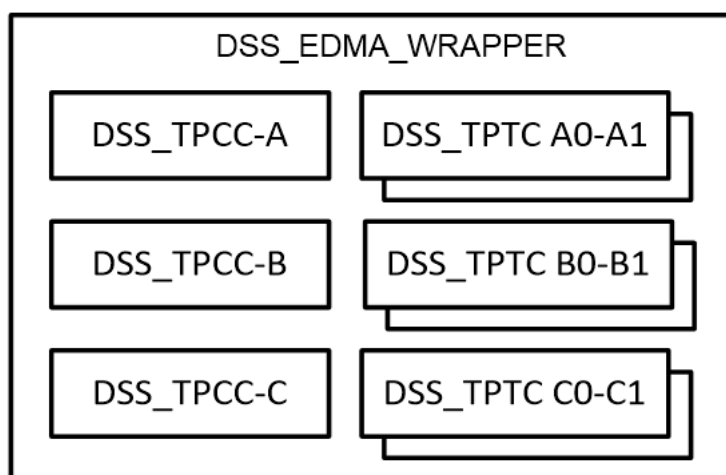
**Table 11-1. MSS TPCC Configuration**

Parameters	TPCC_A	TPCC_B
DMA Ch	64	64
Param Entires	128	128
QDMA Ch	8	8
Event queues	2	2
Mem Protection	Yes	Yes
Channel Mapping	Yes	Yes
Num TCs	2	1
Num Int Ch	NA	NA
Num Regions	NA	NA

**Table 11-2. MSS TPTC Configuration**

Parameters	TPTCA0-A1-B0
FIFO Size	512
TR Pipe Depth	2
Bus Width	8
Read Cmd Num	8
Write Cmd Num	8
RAM ECC	1

### 11.2.3 EDMA Integration in DSS



**Figure 11-4. DSS EDMA Wrapper Block Diagram**

The DSS has 3 TPCCs and 6 TPTCs.

[Table 11-3](#) and [Table 11-4](#) list the configuration for the TPCCs and TPTCs in the DSS.

**Table 11-3. DSS TPCC Configuration**

Parameters	TPCC_A	TPCC_B	TPCC_C
DMA Ch	64	64	64
Param Entires	128	128	256
QDMA Ch	8	8	8
Event queues	2	2	6
Mem Protection	Yes	Yes	Yes
Channel Mapping	Yes	Yes	Yes
Num TCs	2	2	2 (6)
Num Int Ch	NA	NA	NA
Num Regions	NA	NA	NA

**Table 11-4. DSS TPTC Configuration**

Parameters	TPTC_A[0-1]-B[0-1]-C[0-1]	TPTC_C[2-5]
FIFO Size	512	128
TR Pipe Depth	2	2
Bus Width	16	16
Read Cmd Num	8	8
Write Cmd Num	8	8

### 11.2.4 EDMA Functional Interrupt Aggregator

The following EDMA interrupts are aggregated and sent to the processor:

- TPCC Completion Interrupt
- TPCC Completion Region Interrupts
- TPTCs Completion Interrupt

[Table 11-5](#) lists the associated interrupt and registers for each TPCC instance. This section refers to TPCC and TPTC related interrupts, so refer to EDMA XBAR INTRTR0 for EDMA XBAR interrupts.



**Table 11-5. TPCC Interrupt Aggregators**

TPCC	Interrupt	Registers Space
MSS_TPCC_A	MSS_TPCC_A_INTAGG	MSS_CTRL::MSS_TPCC_A_INTAGG_MASK MSS_CTRL::MSS_TPCC_A_INTAGG_STATUS MSS_CTRL::MSS_TPCC_A_INTAGG_STATUS_RAW
MSS_TPCC_B	MSS_TPCC_B_INTAGG	MSS_CTRL::MSS_TPCC_B_INTAGG_MASK MSS_CTRL::MSS_TPCC_B_INTAGG_STATUS MSS_CTRL::MSS_TPCC_B_INTAGG_STATUS_RAW
DSS_TPCC_A	DSS_TPCC_A_INTAGG	DSS_CTRL::DSS_TPCC_A_INTAGG_MASK DSS_CTRL::DSS_TPCC_A_INTAGG_STATUS DSS_CTRL::DSS_TPCC_A_INTAGG_STATUS_RAW
DSS_TPCC_B	DSS_TPCC_B_INTAGG	DSS_CTRL::DSS_TPCC_B_INTAGG_MASK DSS_CTRL::DSS_TPCC_B_INTAGG_STATUS DSS_CTRL::DSS_TPCC_B_INTAGG_STATUS_RAW
DSS_TPCC_C	DSS_TPCC_C_INTAGG	DSS_CTRL::DSS_TPCC_C_INTAGG_MASK DSS_CTRL::DSS_TPCC_C_INTAGG_STATUS DSS_CTRL::DSS_TPCC_C_INTAGG_STATUS_RAW
RSS_TPCC_A	RSS_TPCC_A_INTAGG	RSS_CTRL::RSS_TPCC_A_INTAGG_MASK RSS_CTRL::RSS_TPCC_A_INTAGG_STATUS RSS_CTRL::RSS_TPCC_A_INTAGG_STATUS_RAW
HSM_TPCC_A	HSM_TPCC_A_INTAGG	HSM_CTRL::HSM_TPCC_A_INTAGG_MASK HSM_CTRL::HSM_TPCC_A_INTAGG_STATUS HSM_CTRL::HSM_TPCC_A_INTAGG_STATUS_RAW

For an event to generate an interrupt to the processor, the corresponding bit field must be unmasked in TPCC\_x\_INTAGG\_MASK.

An interrupt processor can read the TPCC\_x\_INTAGG\_STATUS register to detect which event triggered the interrupt.

The interrupt can be cleared by writing 0x1 to the corresponding bit in TPCC\_x\_INTAGG\_STATUS. The software must verify that all the aggregated interrupts are cleared so that the level interrupt is deserted before exiting the ISR. This makes sure that a new pulse interrupt is generated to the processor. Thus, after clearing software, the user can read the register to confirm a value of 0x0.

The register TPCC\_x\_INTAGG\_STATUS\_RAW is set on an event irrespective of the value in TPCC\_x\_INTAGG\_MASK. This field can be cleared by writing 0x1 to the corresponding bit in TPCC\_x\_INTAGG\_STATUS\_RAW.

### 11.2.5 EDMA Error Interrupt Aggregator

The following interrupts are aggregated and sent to the processor:

- TPCC Error
- TPCC MPU Error
- TPTCs Error
- TPCC Read and Write Config Space Access error
- TPTCs Read and Write Config Space Access error

**Table 11-6. TPCC Error Interrupt Aggregators**

TPCC	Interrupt	Registers Space
MSS_TPCC_A	MSS_TPCC_A_ERRAGG	MSS_CTRL::MSS_TPCC_A_ERRAGG_MASK MSS_CTRL::MSS_TPCC_A_ERRAGG_STATUS MSS_CTRL::MSS_TPCC_A_ERRAGG_STATUS_RAW

**Table 11-6. TPCC Error Interrupt Aggregators (continued)**

TPCC	Interrupt	Registers Space
MSS_TPCC_B	MSS_TPCC_B_ERRAGG	MSS_CTRL::MSS_TPCC_B_ERRAGG_MASK MSS_CTRL::MSS_TPCC_B_ERRAGG_STATUS MSS_CTRL::MSS_TPCC_B_ERRAGG_STATUS_RAW
DSS_TPCC_A	DSS_TPCC_A_ERRAGG	DSS_CTRL::DSS_TPCC_A_ERRAGG_MASK DSS_CTRL::DSS_TPCC_A_ERRAGG_STATUS DSS_CTRL::DSS_TPCC_A_ERRAGG_STATUS_RAW
DSS_TPCC_B	DSS_TPCC_B_ERRAGG	DSS_CTRL::DSS_TPCC_B_ERRAGG_MASK DSS_CTRL::DSS_TPCC_B_ERRAGG_STATUS DSS_CTRL::DSS_TPCC_B_ERRAGG_STATUS_RAW
DSS_TPCC_C	DSS_TPCC_C_ERRAGG	DSS_CTRL::DSS_TPCC_C_ERRAGG_MASK DSS_CTRL::DSS_TPCC_C_ERRAGG_STATUS DSS_CTRL::DSS_TPCC_C_ERRAGG_STATUS_RAW
RSS_TPCC_A	RSS_TPCC_A_ERRAGG	RSS_CTRL::RSS_TPCC_A_ERRAGG_MASK RSS_CTRL::RSS_TPCC_A_ERRAGG_STATUS RSS_CTRL::RSS_TPCC_A_ERRAGG_STATUS_RAW
HSM_TPCC_A	HSM_TPCC_A_ERRAGG	HSM_CTRL::HSM_TPCC_A_ERRAGG_MASK HSM_CTRL::HSM_TPCC_A_ERRAGG_STATUS HSM_CTRL::HSM_TPCC_A_ERRAGG_STATUS_RAW

For an event to generate an interrupt to the processor, the corresponding bit field must be unmasked in TPCC\_x\_ERRAGG\_MASK.

An interrupt processor can read the TPCC\_x\_ERRAGG\_STATUS register to detect which event triggered the interrupt.

The interrupt can be cleared by writing 0x1 to the corresponding bit in TPCC\_x\_ERRAGG\_STATUS. The software must ensure that all the aggregated interrupts are cleared so that the level interrupt is deserted before exiting the ISR. Only then is it ensured that a new pulse interrupt is generated to the processor. Thus, after clearing the software, the user should read the register to confirm a value of 0x0.

The register TPCC\_x\_ERRAGG\_STATUS\_RAW is set on an event irrespective of the value in TPCC\_x\_ERRAGG\_MASK. This field can be cleared by writing 0x1 to the corresponding bit in TPCC\_x\_ERRAGG\_STATUS\_RAW.

### 11.2.6 EDMA Configuration

- The MSS has 2 channel controllers: MSS\_TPCC-A and MSS\_TPCC-B. MSS\_TPCC-A has two transfer controllers: MSS\_TPTC-A0 and MSS\_TPTC-A1. MSS\_TPCC-B has one transfer controller MSS\_TPTC-B0.
- The DSS has 3 channel controllers: DSS\_TPCC-A, DSS\_TPCC-B and DSS\_TPCC-C. Each channel controller has 2 transfer controllers.
- RSS has 1 channel controller: RCSS\_TPCC-A, it has two transfer controllers: RCSS\_TPTC-A0 and RCSS\_TPTC-A1.

**Table 11-7. EDMA3 Channel Controller Configuration**

Parameters	MSS_TPCC_A	MSS_TPCC_B	DSS_TPCC_A	DSS_TPCC_B	DSS_TPCC_C	RCSS_TPCC_A
DMA Channel	64	64	64	64	64	64
Param Entires	128	128	128	128	256	128
QDMA Channel	8	8	8	8	8	8
Event queues	2	2	2	2		2
Mem Protection	Yes	Yes	Yes	Yes	Yes	Yes

**Table 11-7. EDMA3 Channel Controller Configuration (continued)**

Parameters	MSS_TPCC_A	MSS_TPCC_B	DSS_TPCC_A	DSS_TPCC_B	DSS_TPCC_C	RCSS_TPCC_A
Channel Mapping	Yes	Yes	Yes	Yes	Yes	Yes
Num TCs	2	2	2	2		2
Num Interrupt Channel	64	64	64	64	64	64
Num Regions	8	8	8	8	8	8

**Table 11-8. EDMA3 Transfer Controller Configuration**

Parameters	MSS_TPTCA0-A1-B0	DSS_TPTC-A[0-1]-B[0-1]-C[0-1]	RCSS_TPTC-A[0-1]
FIFO Size	512	512	512
TR Pipe Depth	2	2	2
Bus Width	8	16	8
Read Cmd Num	8	8	8
Write Cmd Num	8	8	8
RAM ECC	1	Yes	Yes

### Default Burst Size configuration (DBS)

All TPTC in the supports four different default-burst-sizes which are configurable. [Table 11-9](#) shows the config-value to DBS mapping.

**Table 11-9. Config Value to DBS Mapping**

Config value	Burst size
2'b00	16 bytes
2'b01	32 bytes
2'b10	64 bytes
2'b11	128 bytes

**Table 11-10. TPTC DBS Configuration Registers**

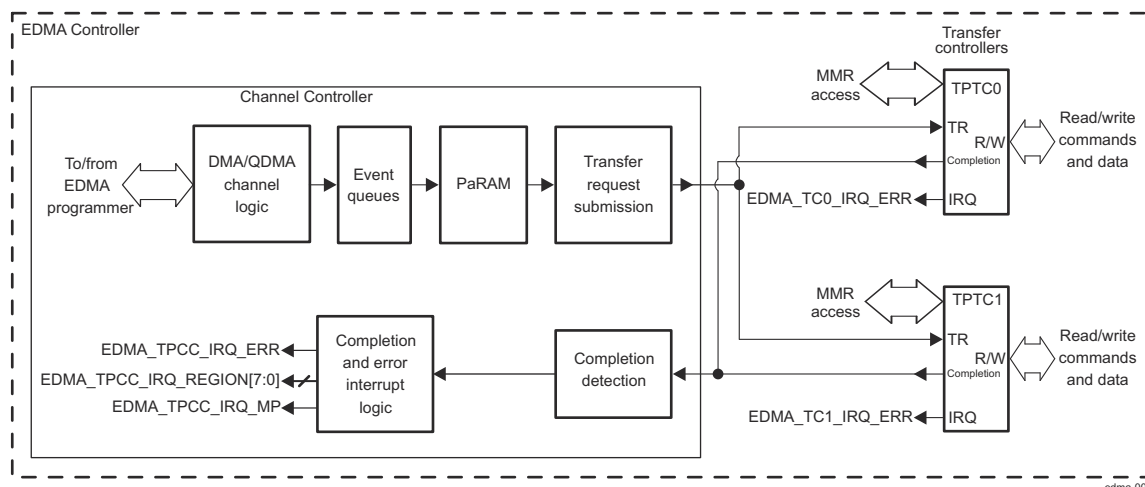
TPTC instance	Corresponding Register
MSS_TPTC_A<0/1>	MSS_CTRL::TPTC_DBS_CONFIG::TPTC_DBS_CONFIG_TPTC_A<0/1>
MSS_TPTC_B0	MSS_CTRL::TPTC_DBS_CONFIG::TPTC_DBS_CONFIG_TPTC_B0
DSS_TPTC_A<0/1>	DSS_CTRL::TPTC_DBS_CONFIG::TPTC_DBS_CONFIG_TPTC_A<0/1>
DSS_TPTC_B<0/1>	DSS_CTRL::TPTC_DBS_CONFIG::TPTC_DBS_CONFIG_TPTC_B<0/1>
DSS_TPTC_C<0-5>	DSS_CTRL::TPTC_DBS_CONFIG::TPTC_DBS_CONFIG_TPTC_C<0-5>
RCSS_TPTC_A<0/1>	RCSS_CTRL::TPTC_DBS_CONFIG::TPTC_DBS_CONFIG_TPTC_A<0/1>

## 11.3 EDMA Controller Functional Description

This chapter discusses the architecture of the EDMA controller. The description contained in this section is generic to the EDMA module, and not all features mentioned here are supported by the device. See the EDMA integration section of the device to determine the applicability of these features.

### 11.3.1 Block Diagram

Figure 11-5 shows the functional block diagram of the EDMA controller.



**Figure 11-5. EDMA Controller Block Diagram**

#### 11.3.1.1 Third-Party Channel Controller

The TPCC is the EDMA transfer scheduler responsible for scheduling, arbitrating, and issuing user programmed transfers to the two TPTCs.

The functional block diagram below describes EDMA channel controller (EDMA\_TPCC).

The main blocks of the EDMA\_TPCC are as follows:

- **Parameter RAM (PaRAM):** The PaRAM maintains parameter sets for channel and reload parameter sets. The PaRAM must be written with the transfer context for the desired channels and link parameter sets. EDMA\_TPCC processes and sets based on a trigger event and submits a transfer request (TR) to the transfer controllers.
- **EDMA event and interrupt processing registers:** Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
- **Completion detection:** The completion detect block detects completion of transfers by the EDMA\_TPTCs or follower peripherals. The completion of transfers can be used optionally to chain trigger new transfers or to assert interrupts.
- **Event queues:** Event queues form the interface between the event detection logic and the transfer request submission logic.
- **Memory protection registers:** Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

Other functions include the following:

- **Region registers:** Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA programmers own (for example, DSPs).
- **Debug registers:** Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA\_TPCC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels).

Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. These channels are identical. The main difference between a DMA channel and a QDMA channel is the method that the system uses to trigger transfers.

- DMA channels are triggered by external events by the event set registers EDMA\_TPCC\_ESR and EDMA\_TPCC\_ESRH, or through chaining register EDMA\_TPCC\_CER.
- QDMA channels are triggered automatically (auto-triggered) by the CPU. QDMAs allow a minimum number of linear writes to be issued to the TPCC to force a series of transfers to occur.

The TPCC arbitrates among pending DMA and QDMA events with a fixed [64:1] and [8:1] priority encoder for these events, respectively (a low channel number corresponds to a high priority).

DMA events are always higher priority than QDMA events. The higher-priority event is placed in the event queue to await submission to the transfer controllers, which occurs at the earliest opportunity. Each event queue is serviced in FIFO order, with a maximum of 16 queued events per event queue. If more than one TPTC is ready to be programmed with a transmission request (TR), the event queues are serviced with fixed priority: Q0 is higher than Q1. When an event is ready to be queued and the event queue and the TC channel are empty, the event bypasses the event queue and goes directly to the PaRAM processing logic for submission to the appropriate TC. If the transfer request TR bus or PaRAM processing are busy, the bypass path is not used. The bypass is not used to dequeue for a higher-priority event.

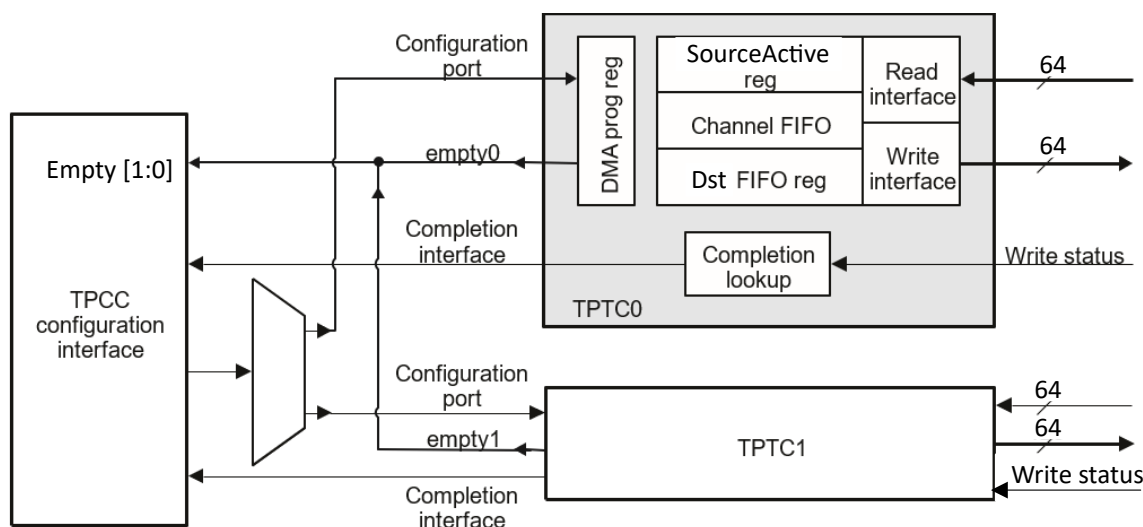
Events are extracted from the event queue when the EDMA\_TPTC is available for a new TR to be programmed into the EDMA\_TPTC (signaled with the empty signal, indicating an empty program register set). As an event is extracted from the event queue, the associated PaRAM entry is processed and submitted to the TPTC as a TR. The TPCC updates the appropriate counts and addresses in the PaRAM entry in anticipation of the next trigger event for that PaRAM entry.

The EDMA\_TPCC also has an error detection logic that causes an error interrupt generation on various error conditions (for example: missed events EDMA\_TPCC\_EMR and EDMA\_TPCC\_EMRH registers, exceeding event queue thresholds in EDMA\_TPCC\_CCERR register, etc.).

#### 11.3.1.2 Third-Party Transfer Controller

The TPTC module is the EDMA transfer engine that generates transfers as programmed in dedicated working registers, using two dedicated controller ports: a read-only port and a write-only port.

[Figure 11-6](#) shows a functional block diagram and of the EDMA transfer controller (EDMA\_TPTC) and its connection to the EDMA\_TPCC.



**Figure 11-6. TPTC Block Diagram**

#### Note

The port data bus width of the instances of the TPTC is fixed at 64 bits.

Two instances of the EDMA\_TPTC generate concurrent traffic on the L3\_VBUSM interconnect. Each TC controller consists of the following components:

- **DMA Program Register Set:** Stores the context for the DMA transfer that is loaded into the active register set when the current active register set completes. The CPU or TPCC programs the Program Register Set, not the active register set. For typical standalone operation, the CPU programs the Program Register while the TC services the Active register set. The Program Register set includes ownership control such that CPU software and the EDMA stay synchronized relative to one another.
- **Source Active Register Set :** Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in progress in the Read Controller. The Active register set is split into independent Source and Destination, because the source interconnect controller and the destination interconnect controller operate independently of one another.
- **Destination FIFO Register Set:** Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in progress, or pending, in the Write Controller. The pending register must allow the source controller to begin processing a new TR while the destination register set processes the previous TR.
- **Channel FIFO:** Temporary holding buffer for in-flight data. The read return data of the source peripheral is stored in the Data FIFO, and then is written to the destination peripheral by the write command/data bus.
- **Read Controller/Interconnect Read Interface:** The Interconnect read interface issues optimally sized read commands to the source peripheral, based on a burst size of 32 bytes and available landing space in the channel FIFO.
- **Write controller/Interconnect Write interface:** The local interconnect write interface issues optimally sized write commands to the destination peripheral, based on a burst size of 32 bytes and available data in the channel FIFO.
- **Completion interface:** sends completion codes to the EDMA\_TPCC when a transfer completes and generates interrupts and chained events in the TPCC module.
- **Configuration port:** Target interface that provides read/write access to program registers and read access to all memory-mapped TPTC registers.

When one EDMA\_TPTC module is idle and receive its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second

TR (if pending from EDMA\_TPCC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

The read controller issues read commands controlled by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

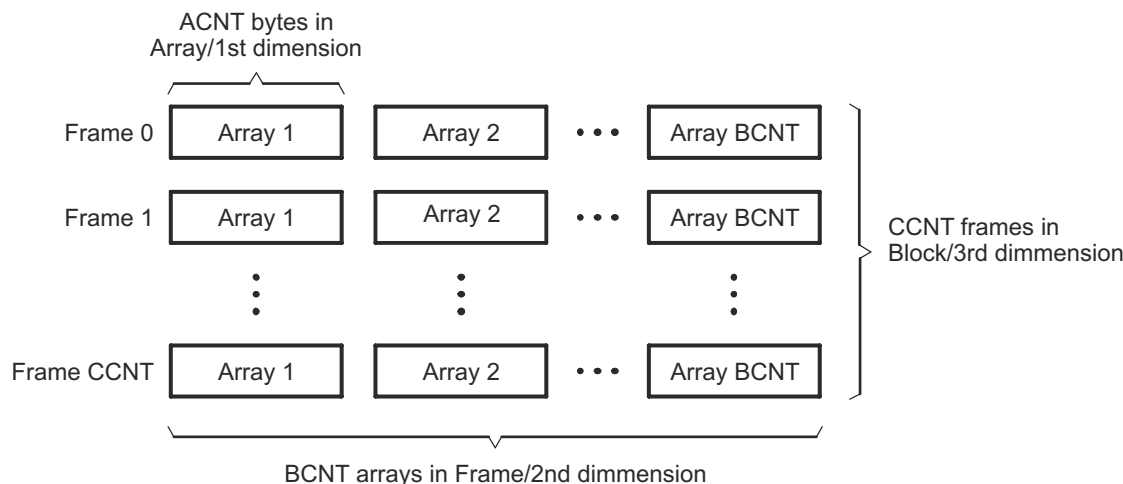
### 11.3.2 Types of EDMA Controller Transfers

An EDMA transfer is always defined in terms of three dimensions. Figure 11-7 shows the three dimensions used by EDMA controller transfers. These three dimensions are defined as:

- 1st Dimension or Array (A): The 1st dimension in a transfer consists of EDMA\_TPCC\_ABCNT\_n[15:0] ACNT contiguous bytes.
- 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of EDMA\_TPCC\_ABCNT\_n[31:16] BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using bit-fields EDMA\_TPCC\_BIDX\_n[15:0] SBIDX or EDMA\_TPCC\_BIDX\_n[31:16] DBIDX.
- 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. The Count for 3rd Dimension is defined in PaRAM memory EDMA\_TPCC\_CCNT\_n[15:0] CCNT. Each transfer in the 3rd dimension is separated from the previous by an index programmed using EDMA\_TPCC\_CIDX\_n[15:0] SCIDX or EDMA\_TPCC\_CIDX\_n[31:16] DCIDX.

#### Note

The reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (EDMA\_TPCC\_OPT\_n[2] SYNCDIM bit). For these three dimensions, only two synchronization types are supported: A-synchronized transfers and AB-synchronized transfers.



edma-007

**Figure 11-7. Definition of ACNT, BCNT, and CCNT**

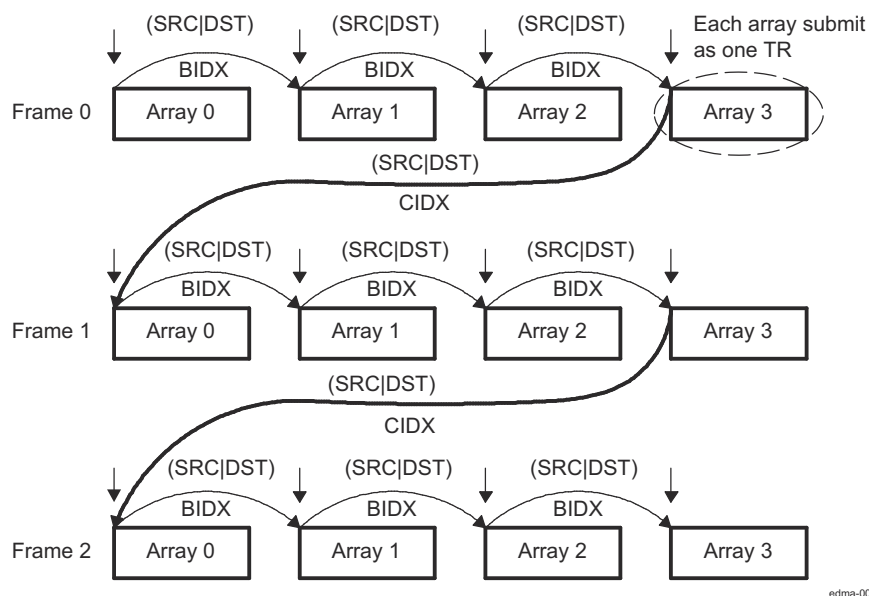
### 11.3.2.1 A-Synchronized Transfers

In an A-synchronized transfer, each EDMA sync event initiates the transfer of the 1st dimension of EDMA\_TPCC\_ABCNT\_n[15:0] ACNT bytes, or one array of ACNT bytes. Each event/TR packet conveys the transfer information for one array only. Thus, BCNT × CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA\_TPCC\_BIDX\_n[15:0] SBIDX and EDMA\_TPCC\_BIDX\_n[31:16] DBIDX, as shown in Figure 11-8, where the start address of Array N is equal to the start address of Array N – 1 plus source (SRC) or destination (DST) in EDMA\_TPCC\_BIDX\_n register.

Frames are always separated by EDMA\_TPCC\_CIDX\_n[15:0] SCIDX and EDMA\_TPCC\_CIDX\_n[31:16] DCIDX. For A-synchronized transfers, after the frame is exhausted, the address is updated by adding SRCCIDX/ DSTCIDX to the beginning address of the last array in the frame. As in Figure 11-8, SRCCIDX / DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 11-8 shows an A-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 12 sync events (BCNT × CCNT) exhaust a PaRAM set. See Figure 11-8 for details on parameter set updates.



**Figure 11-8. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)**



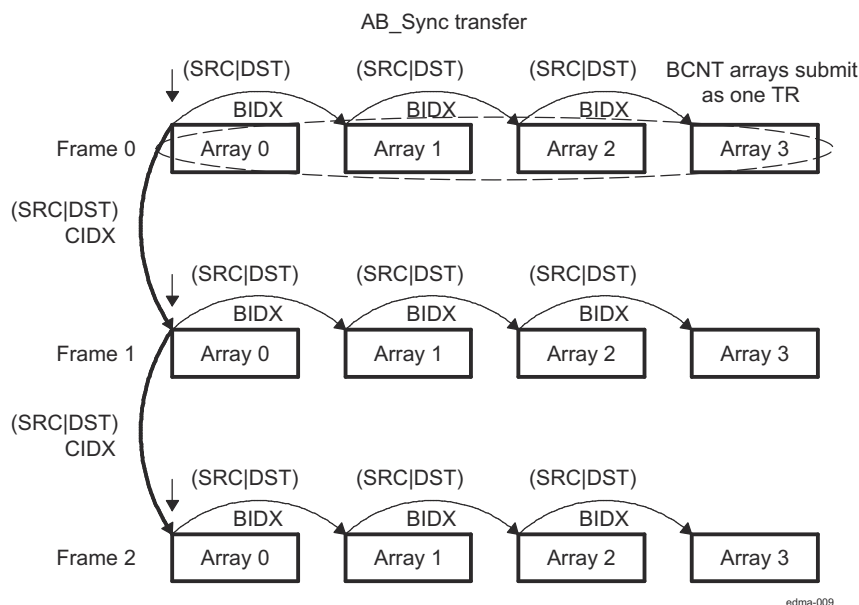
### 11.3.2.2 AB-Synchronized Transfers

In a AB-synchronized transfer, each EDMA sync event initiates the transfer of 2 dimensions or one frame. Each event/TR packet conveys information for one entire frame of BCNT\_n arrays of ACNT\_n bytes. Thus, EDMA\_TPCC\_CCNT\_n events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA\_TPCC\_BIDX\_n[15:0] SBIDX and EDMA\_TPCC\_BIDX\_n[31:16] DBIDX as shown in Figure 11-9. Frames are always separated by SRCCIDX and DSTCIDX.

Note that for AB-synchronized transfers, after a TR for the frame is submitted, the address update is to add EDMA\_TPCC\_CIDX\_n[15:0] SCIDX / EDMA\_TPCC\_CIDX\_n[31:16] DCIDX to the beginning address of the beginning array in the frame. This is different from A-synchronized transfers where the address is updated by adding SRCCIDX/DSTCIDX to the start address of the last array in the frame. See Section 11.3.3.6 for details on parameter set updates.

Figure 11-9 shows an AB-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 3 sync events (CCNT) exhaust a PaRAM set; that is, a total of 3 transfers of 4 arrays each completes the transfer.



**Figure 11-9. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)**

#### Note

ABC-synchronized transfers are not directly supported. It can be logically achieved by chaining between multiple AB-synchronized transfers.

### 11.3.3 Parameter RAM (PaRAM)

The EDMA controller is a RAM-based architecture. The transfer context (source/destination addresses, count, indexes, etc.) for DMA or QDMA channels is programmed in a parameter RAM table in EDMA\_TPCC. The PaRAM table is segmented into multiple PaRAM sets. Each PaRAM set includes eight four-byte PaRAM set entries (32-bytes total per PaRAM set), which includes typical DMA transfer parameters such as source address, destination address, transfer counts, indexes, options, etc.

The PaRAM structure supports flexible ping-pong, circular buffering, channel chaining, and auto-reloading (linking).

The contents of the PaRAM include the following:

- PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels
- 8 channels remain for link or QDMA sets

By default, all channels map to PaRAM set to 0 and should be remapped before use by EDMA\_TPCC\_DCHMAPN\_m and EDMA\_TPCC\_QCHMAPN\_j registers.

**Table 11-11. EDMA Parameter RAM Contents**

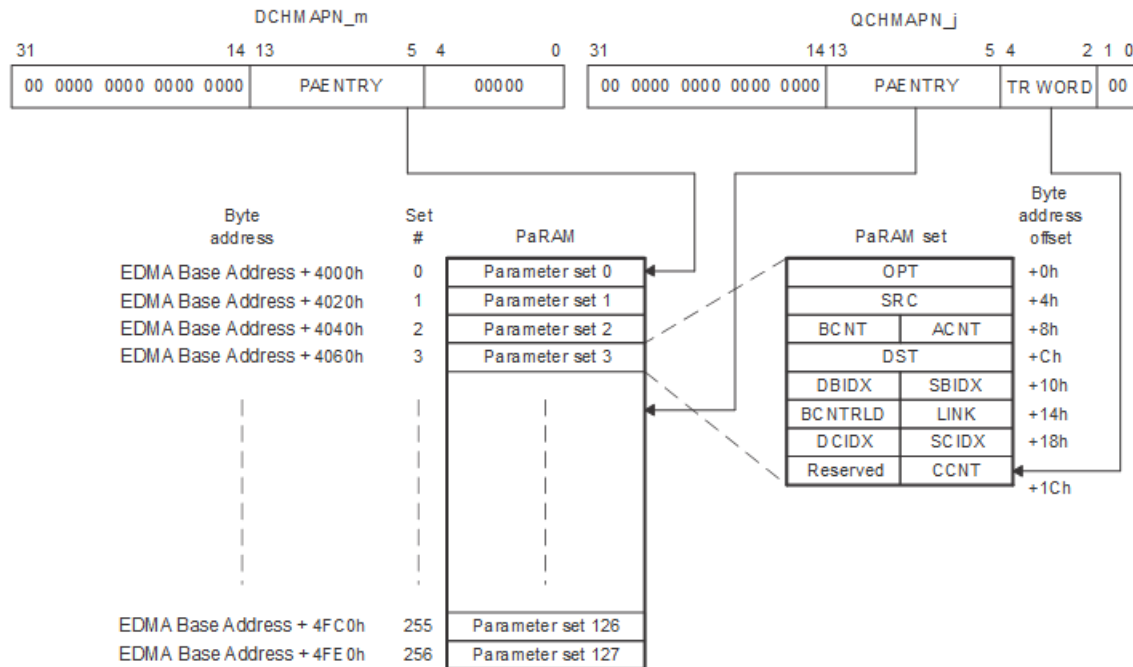
PaRAM Set Number	Base Address	Parameters <sup>(1)</sup>
0	EDMA Base Address + 4000h to EDMA Base Address + 401Fh	PaRAM set 0
1	EDMA Base Address + 4020h to EDMA Base Address + 403Fh	PaRAM set 1
2	EDMA Base Address + 4040h to EDMA Base Address + 405Fh	PaRAM set 2
3	EDMA Base Address + 4060h to EDMA Base Address + 407Fh	PaRAM set 3
4	EDMA Base Address + 4080h to EDMA Base Address + 409Fh	PaRAM set 4
5	EDMA Base Address + 40A0h to EDMA Base Address + 40BFh	PaRAM set 5
6	EDMA Base Address + 40C0h to EDMA Base Address + 40DFh	PaRAM set 6
7	EDMA Base Address + 40E0h to EDMA Base Address + 40FFh	PaRAM set 7
8	EDMA Base Address + 4100h to EDMA Base Address + 411Fh	PaRAM set 8
9	EDMA Base Address + 4120h to EDMA Base Address + 413Fh	PaRAM set 9
...	...	...
63	EDMA Base Address + 47E0h to EDMA Base Address + 47FFh	PaRAM set 63
64	EDMA Base Address + 4800h to EDMA Base Address + 481Fh	PaRAM set 64
65	EDMA Base Address + 4820h to EDMA Base Address + 483Fh	PaRAM set 65
...	...	...
127	EDMA Base Address + 5000h to EDMA Base Address + 4FE0h	PaRAM set 127

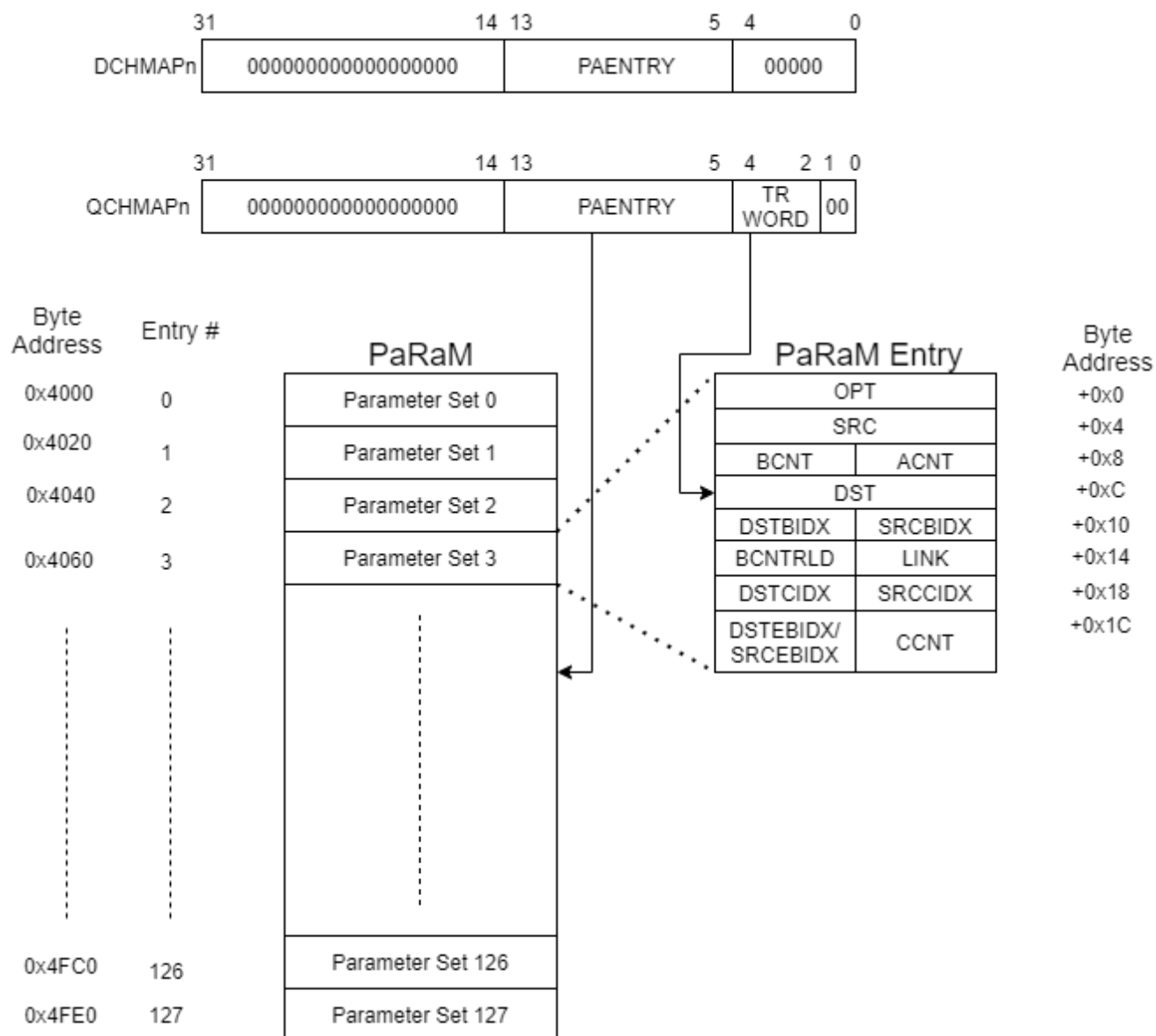
(1) The device has 8 QDMA channels that can be mapped to any parameter set number from 0 to .

### Note

#### 11.3.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in [Figure 11-10](#) and described in [Table 11-12](#). Each PaRAM set consists of 16-bit and 32-bit parameters.




**Figure 11-10. PaRAM Set**

### Note

Figure above is a representation of 128 bit entries. For device specific details please refer to [EDMA Configuration](#) chapter.

**Table 11-12. EDMA Channel Parameter Description**

Offset Address (bytes)	Acronym	Parameter	Description
0h	OPT	Channel Options EDMA_TPCC_OPT_n register	Transfer configuration options
4h	SRC	Channel Source Address EDMA_TPCC_SRC_n register	The byte address from which data is transferred
8h <sup>(1)</sup>	ACNT	Count for 1st Dimension EDMA_TPCC_ABCNT_n[15:0] ACNT bit-field.	Unsigned value specifying the number of contiguous bytes within an array (first dimension of the transfer). Valid values range from 1 to 65 535.
	BCNT	Count for 2nd Dimension EDMA_TPCC_ABCNT_n[31:16] BCNT bit-field.	Unsigned value specifying the number of arrays in a frame, where an array is ACNT bytes. Valid values range from 1 to 65 535.
Ch	DST	Channel Destination Address EDMA_TPCC_DST_n register	The byte address to which data is transferred
10h <sup>(1)</sup>	SBIDX	Source BCNT Index EDMA_TPCC_BIDX_n[15:0] SBIDX bit-field.	Signed value specifying the byte address offset between source arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
	DBIDX	Destination BCNT Index EDMA_TPCC_BIDX_n[31:16] DBIDX bit-field.	Signed value specifying the byte address offset between destination arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
14h <sup>(1)</sup>	LINK	Link Address EDMA_TPCC_LNK_n[15:0] LINK bit-field	The PaRAM address containing the PaPARAM set to be linked (copied from) when the current PaPARAM set is exhausted. A value of FFFFh specifies a null link.
	BCNTRLD	BCNT Reload EDMA_TPCC_LNK_n[31:16] BCNTRLD bit-field	The count value used to reload BCNT when BCNT decrements to 0 (TR is submitted for the last array in 2nd dimension). Only relevant in A-synchronized transfers.
18h <sup>(1)</sup>	SCIDX	Source CCNT index. EDMA_TPCC_CIDX_n[15:0] SCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767.  A-synchronized transfers: The byte address offset from the beginning of the last source array in a frame to the beginning of the first source array in the next frame.  AB-synchronized transfers: The byte address offset from the beginning of the first source array in a frame to the beginning of the first source array in the next frame.
	DCIDX	Destination CCNT index. EDMA_TPCC_CIDX_n[31:16] DCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767.  A-synchronized transfers: The byte address offset from the beginning of the last destination array in a frame to the beginning of the first destination array in the next frame.  AB-synchronized transfers: The byte address offset from the beginning of the first destination array in a frame to the beginning of the first destination array in the next frame.
1Ch	CCNT	Count for 3rd Dimension. EDMA_TPCC_CCNT_n[15:0] CCNT bit-field.	Unsigned value specifying the number of frames in a block, where a frame is BCNT arrays of ACNT bytes. Valid values range from 1 to 65 535.
	Reserved	Reserved	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts can result in undefined behavior.

- (1) If OPT, SRC, or DST is the trigger word for a QDMA transfer, then it is required to do a 32-bit access to that field. Furthermore, it is recommended to perform only 32-bit accesses on the parameter RAM for best code compatibility. For example, switching the endianness of the processor swaps addresses of the 16-bit fields, but 32-bit accesses avoid the issue entirely.

### 11.3.3.2 EDMA Channel PaRAM Set Entry Fields

#### 11.3.3.2.1 Channel Options Parameter (OPT)

This is the control register for TPCC channel configuration options. Refer to the EDMA\_TPCC\_OPT\_n register bitfield description in the for additional details.

#### 11.3.3.2.2 Channel Source Address (SRC)

The 32-bit source address parameter specifies the starting byte address of the source. For SAM in increment mode, there are no alignment restrictions imposed by EDMA. For SAM in FIFO addressing mode, it must program the source address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA\_TPTC returns an error. Refer to [Section 11.3.12.3 Error Generation](#) for additional details.

#### 11.3.3.2.3 Channel Destination Address (DST)

The 32-bit destination address parameter specifies the starting byte address of the destination. For DAM in increment mode, there are no alignment restrictions imposed by EDMA. For DAM in FIFO addressing mode, it must program the destination address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA\_TPTC returns an error. Refer to *Error Generation* for additional details.

#### 11.3.3.2.4 Count for 1st Dimension (ACNT)

EDMA\_TPCC\_ABCNT\_n[15:0] ACNT represents the number of bytes within the 1st dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 1 and 65535. Therefore, the maximum number of bytes in an array is 65 535 bytes (64K – 1 bytes). ACNT must be greater than or equal to 1 for a TR to be submitted to EDMA\_TPTC. A transfer with ACNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA\_TPCC\_OPT\_n.

Refer to [Section 11.3.3.5 Dummy Versus Null Transfer Comparison](#) and [Section 11.3.5.3 Dummy or Null Completion](#) for details on dummy/null completion conditions.

#### 11.3.3.2.5 Count for 2nd Dimension (BCNT)

EDMA\_TPCC\_ABCNT\_n[15:0] BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT are between 1 and 65 535. Therefore, the maximum number of arrays in a frame is 65 535 (64K – 1 arrays). A transfer with BCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA\_TPCC\_OPT\_n.

Refer to [Section 11.3.3.5 Dummy Versus Null Transfer Comparison](#) and [Section 11.3.5.3 Dummy or Null Completion](#) for details on dummy/null completion conditions.

#### 11.3.3.2.6 Count for 3rd Dimension (CCNT)

EDMA\_TPCC\_CCNT\_n[15:0] CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT are between 1 and 65 535. Therefore, the maximum number of frames in a block is 65 535 (64K – 1 frames). A transfer with CCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA\_TPCC\_OPT\_n.

A CCNT value of 0 is considered either a null or dummy transfer.

Refer to [Section 11.3.3.5 Dummy Versus Null Transfer Comparison](#) and [Section 11.3.5.3 Dummy or Null Completion](#) for details on dummy/null completion conditions.

#### 11.3.3.2.7 BCNT Reload (BCNTRLD)

EDMA\_TPCC\_LNK\_n[31:16] BCNTRLD is a 16-bit unsigned value used to reload the EDMA\_TPCC\_ABCNT\_n[15:0] BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-synchronized transfers. In this case, the EDMA\_TPCC decrements the BCNT value by 1 on

each TR submission. When BCNT reaches 0, the EDMA\_TPCC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value.

For AB-synchronized transfers, the EDMA\_TPCC submits the BCNT in the TR and the EDMA\_TPTC decrements BCNT appropriately. For AB-synchronized transfers, BCNTRLD is not used.

#### 11.3.3.2.8 Source B Index (SBIDX)

EDMA\_TPCC\_BIDX\_n[15:0] SBIDX is a 16-bit signed value (2s complement) used for source address modification between each array in the 2nd dimension. Valid values for EDMA\_TPCC\_BIDX\_n[15:0] SBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-synchronized and AB-synchronized transfers. Some examples:

- EDMA\_TPCC\_BIDX\_n[15:0] SBIDX = 0000h (0): no address offset from the beginning of an array to the beginning of the next array. All arrays are fixed to the same beginning address.
- EDMA\_TPCC\_BIDX\_n[15:0] SBIDX = 0003h (+3): the address offset from the beginning of an array to the beginning of the next array in a frame is 3 bytes. For example, if the current array begins at address 1000h, the next array begins at 1003h.
- EDMA\_TPCC\_BIDX\_n[15:0] SBIDX = FFFFh (–1): the address offset from the beginning of an array to the beginning of the next array in a frame is –1 byte. For example, if the current array begins at address 5054h, the next array begins at 5053h.

#### 11.3.3.2.9 Destination B Index (DBIDX)

EDMA\_TPCC\_BIDX\_n[31:16] DBIDX is a 16-bit signed value (2s complement) used for destination address modification between each array in the 2nd dimension. Valid values for EDMA\_TPCC\_BIDX\_n[31:16] DBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-synchronized and AB-synchronized transfers. Refer to [Section 11.3.3.2.8 Source B Index \(SBIDX\)](#) for examples.

#### 11.3.3.2.10 Source C Index (SCIDX)

EDMA\_TPCC\_CIDX\_n[15:0] SCIDX is a 16-bit signed value (2s complement) used for source address modification in the 3rd dimension. Valid values for EDMA\_TPCC\_CIDX\_n[15:0] SCIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

#### Note

When SCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame ([Figure 11-8](#)), while the current array in an AB-synchronized transfer is the first array in the frame ([Figure 11-9](#)).

#### 11.3.3.2.11 Destination C Index (DCIDX)

EDMA\_TPCC\_CIDX\_n[31:16] DCIDX is a 16-bit signed value (2s complement) used for destination address modification in the 3rd dimension. Valid values are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array TR in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

#### Note

When DCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame ([Figure 11-8](#)), while the current array in a AB-synchronized transfer is the first array in the frame ([Figure 11-9](#)).



### 11.3.3.2.12 Link Address (LINK)

The EDMA\_TPCC provides a mechanism, called linking, to reload the current PaRAM set upon its natural termination (that is, after the count fields are decremented to 0) with a new PaRAM set. The 16-bit parameter EDMA\_TPCC\_LNK\_n[15:0] LINK specifies the byte address offset in the PaRAM from which the EDMA\_TPCC loads/reloads the next PaRAM set during linking.

It must program the link address to point to a valid aligned 32-byte PaRAM set. The 5 LSBs of the LINK field should be cleared to 0.

The EDMA\_TPCC ignores the upper 2 bits of the LINK entry, allowing the flexibility of programming the link address as either an absolute/literal byte address or use the PaRAM-base-relative offset address. Therefore, if it use the literal address with a range from 4000h to 7FFFh, it will be treated as a PaRAM-base-relative value of 0000h to 3FFFh.

It should check that the programmed value in the EDMA\_TPCC\_LNK\_n[15:0] LINK field is correctly, so that link update is requested from a PaRAM address that falls in the range of the available PaRAM addresses on the device.

Value of FFFFh in EDMA\_TPCC\_LNK\_n[15:0] LINK bit-field is referred to as a NULL link that should cause the EDMA\_TPCC to perform an internal write of 0 to all entries of the current PaRAM set, except for the EDMA\_TPCC\_LNK\_n[15:0] LINK field is set to FFFFh. Also, see [Section 11.3.5 Completion of a DMA Transfer](#) for details on terminating a transfer.

### 11.3.3.3 Null PaRAM Set

A null PaRAM set is defined as a PaRAM set where all count fields (EDMA\_TPCC\_ABCNT\_n[15:0] ACNT, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT, and EDMA\_TPCC\_CCNT\_n[15:0] CCNT) are cleared to 0. If a PaRAM set associated with a channel is a NULL set, then when serviced by the EDMA\_TPCC, the bit corresponding to the channel is set in the associated event missed register (EDMA\_TPCC\_EMR, EDMA\_TPCC\_EMRH, or EDMA\_TPCC\_QEMR). This bit remains set in the associated secondary event register (EDMA\_TPCC\_SER, EDMA\_TPCC\_SERH, or EDMA\_TPCC\_QSER).

*This implies that any future events on the same channel are ignored by the EDMA\_TPCC and it is required to clear the bit in EDMA\_TPCC\_SER, EDMA\_TPCC\_SERH, or EDMA\_TPCC\_QSER for the channel. This is considered an error condition, since events are not expected on a channel that is configured as a null transfer.*

### 11.3.3.4 Dummy PaRAM Set

A dummy PaRAM set is defined as a PaRAM set where at least one of the count fields (EDMA\_TPCC\_ABCNT\_n[15:0] ACNT, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT, or EDMA\_TPCC\_CCNT\_n[15:0] CCNT) is cleared to 0 and at least one of the count fields is nonzero.

If a PaRAM set associated with a channel is a dummy set, then when serviced by the EDMA\_TPCC, it will not set the bit corresponding to the channel (DMA/QDMA) in the event missed register (EDMA\_TPCC\_EMR, EDMA\_TPCC\_EMRH, or EDMA\_TPCC\_QEMR) and the secondary event register (EDMA\_TPCC\_SER, EDMA\_TPCC\_SERH, or EDMA\_TPCC\_QSER) bit gets cleared similar to a normal transfer. Future events on that channel are serviced. A dummy transfer is a legal transfer of 0 bytes.

### 11.3.3.5 Dummy Versus Null Transfer Comparison

There are some differences in the way the EDMA\_TPCC logic treats a dummy versus a null transfer request. A null transfer request is an error condition, but a dummy transfer is a legal transfer of 0 bytes. A null transfer causes an error bit (*En*) in EDMA\_TPCC\_EMR to get set and the *En* bit in EDMA\_TPCC\_SER remains set, essentially preventing any further transfers on that channel without clearing the associated error registers.

[Table 11-13](#) summarizes the conditions and effects of null and dummy transfer requests.

**Table 11-13. Dummy and Null Transfer Request**

Feature	Null TR	Dummy TR
EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR is set	Yes	No
EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER remains set	Yes	No



**Table 11-13. Dummy and Null Transfer Request (continued)**

Feature	Null TR	Dummy TR
Link update (STATIC = 0 in EDMA_TPCC_OPT_n)	Yes	Yes
EDMA_TPCC_QER is set	Yes	Yes
EDMA_TPCC_IPR / EDMA_TPCC_IPRH, EDMA_TPCC_CER / EDMA_TPCC_CERH is set using early completion	Yes	Yes

### 11.3.3.6 Parameter Set Updates

When a TR is submitted for a given DMA/QDMA channel and its corresponding PaRAM set, the EDMA\_TPCC is responsible for updating the PaRAM set in anticipation of the next trigger event. For events that are not final, this includes address and count updates; for final events, this includes the link update.

The specific PaRAM set entries that are updated depend on the channel's synchronization type (A-synchronized or AB-synchronized) and the current state of the PaRAM set. A B-update refers to the decrementing of EDMA\_TPCC\_ABCNT\_n[31:16] BCNT in the case of A-synchronized transfers after the submission of successive TRs. A C-update refers to the decrementing of CCNT in the case of A-synchronized transfers after BCNT TRs for EDMA\_TPCC\_ABCNT\_n[15:0] ACNT byte transfers have submitted. For AB-synchronized transfers, a C-update refers to the decrementing of EDMA\_TPCC\_CCNT\_n[15:0] CCNT after submission of every transfer request.

Refer to [Table 11-14](#) for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in [Section 11.3.3.7 Linking Transfers](#).

After the TR is read from the PaRAM (and is in process of being submitted to EDMA\_TPTC), the following fields are updated if needed:

- A-synchronized: BCNT, CCNT, SRC, DST.
- AB-synchronized: CCNT, SRC, DST.

The following fields are not updated (except for during linking, where all fields are overwritten by the link PaRAM set):

- A-synchronized: EDMA\_TPCC\_ABCNT\_n[15:0] ACNT, EDMA\_TPCC\_LNK\_n[31:16] BCNTRLD, EDMA\_TPCC\_BIDX\_n[15:0] SBIDX, EDMA\_TPCC\_BIDX\_n[31:16] DBIDX, EDMA\_TPCC\_CIDX\_n[15:0] SCIDX, EDMA\_TPCC\_CIDX\_n[31:16] DCIDX, EDMA\_TPCC\_OPT\_n, EDMA\_TPCC\_LNK\_n[15:0] LINK.
- AB-synchronized: EDMA\_TPCC\_ABCNT\_n[15:0] ACNT, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT, EDMA\_TPCC\_LNK\_n[31:16] BCNTRLD, EDMA\_TPCC\_BIDX\_n[15:0] SBIDX, EDMA\_TPCC\_BIDX\_n[31:16] DBIDX, EDMA\_TPCC\_CIDX\_n[15:0] SCIDX, EDMA\_TPCC\_CIDX\_n[31:16] DCIDX, EDMA\_TPCC\_OPT\_n, EDMA\_TPCC\_LNK\_n[15:0] LINK.

### Note

PaRAM updates only pertain to the information that is needed to properly submit the next transfer request to the EDMA\_TPTC. Updates that occur while data is moved within a transfer request are tracked within the transfer controller, and is detailed in [Section 11.3.12 EDMA Transfer Controller \(EDMA\\_TPTC\)](#). For A-synchronized transfers, the EDMA\_TPCC always submits a TRP for EDMA\_TPCC\_ABCNT\_n[15:0] ACNT bytes (EDMA\_TPCC\_ABCNT\_n[31:16] BCNT = 1 and EDMA\_TPCC\_CCNT\_n[15:0] CCNT = 1). For AB-synchronized transfers, the EDMA\_TPCC always submits a TRP for EDMA\_TPCC\_ABCNT\_n[15:0] ACNT bytes of BCNT arrays (EDMA\_TPCC\_CCNT\_n[15:0] CCNT = 1). The EDMA\_TPTC is responsible for updating source and destination addresses within the array based on EDMA\_TPCC\_ABCNT\_n[15:0] ACNT and EDMA\_TPCC\_OPT\_n[10:8] FWID. For AB-synchronized transfers, the EDMA\_TPTC is also responsible to update source and destination addresses between arrays based on EDMA\_TPCC\_BIDX\_n[15:0] SBIDX and EDMA\_TPCC\_BIDX\_n[31:16] DBIDX.

[Table 11-14](#) shows the details of parameter updates that occur within EDMA\_TPCC for A-synchronized and AB-synchronized transfers.

**Table 11-14. Parameter Updates in EDMA\_TPCC (for Non-Null, Non-Dummy PaRAM Set)**

	A-Synchronized Transfer			AB-Synchronized Transfer		
	B-Update	C-Update	Link Update	B-Update	C-Update	Link Update
Condition:	BCNT > 1	BCNT == 1 && CCNT > 1	BCNT == 1 && CCNT == 1	N/A	EDMA_TPCC_CCNT_n[15:0] CCNT > 1	EDMA_TPCC_CCNT_n[15:0] CCNT == 1
SRC	+= SBIDX	+= SCIDX	= Link.EDMA_TPCC_SRC_n	in EDMA_TPT C	+= SCIDX	= Link.EDMA_TPCC_SRC_n
DST	+= DBIDX	+= DCIDX	= Link.EDMA_TPCC_DST_n	in EDMA_TPT C	+= DCIDX	= Link.EDMA_TPCC_DST_n
ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT
BCNT	-- 1	= BCNTRL	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT	in EDMA_TPT C	N/A	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT
CCNT	None	-- 1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT	in EDMA_TPT C	-- 1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT
SBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPT C	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
SCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPT C	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0] LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0] LINK
BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD
OPT <sup>(1)</sup>	None	None	= LINK.EDMA_TPCC_OPT_n	None	None	= LINK.EDMA_TPCC_OPT_n

(1) In all cases, no updates occur if EDMA\_TPCC\_OPT\_n[3] STATIC == 1 for the current PaRAM set.

### Note

The EDMA\_TPCC includes no special hardware to detect when an indexed address update calculation overflows/underflows. The address update will wrap across boundaries as programmed by the user. It should ensure that no transfer is allowed to cross internal port boundaries between peripherals. A single TR must target a single source/destination peripheral endpoint.

### 11.3.3.7 Linking Transfers

The EDMA\_TPCC provides a mechanism known as linking, which allows the entire PaRAM set to be reloaded from a location within the PaRAM memory map (for both DMA and QDMA channels). Linking is especially useful for maintaining ping-pong buffers, circular buffering, and repetitive/continuous transfers with no CPU intervention. Upon completion of a transfer, the current transfer parameters are reloaded with the parameter set pointed to by the 16-bit link address field of the current parameter set. Linking only occurs when the EDMA\_TPCC\_OPT\_n[3] STATIC bit is cleared.

---

#### Note

It should always link a transfer (EDMA or QDMA) to another useful transfer. If it must terminate a transfer, then link the transfer to a NULL parameter set. Refer to [Section 11.3.3.3 Null PaRAM Set](#).

---

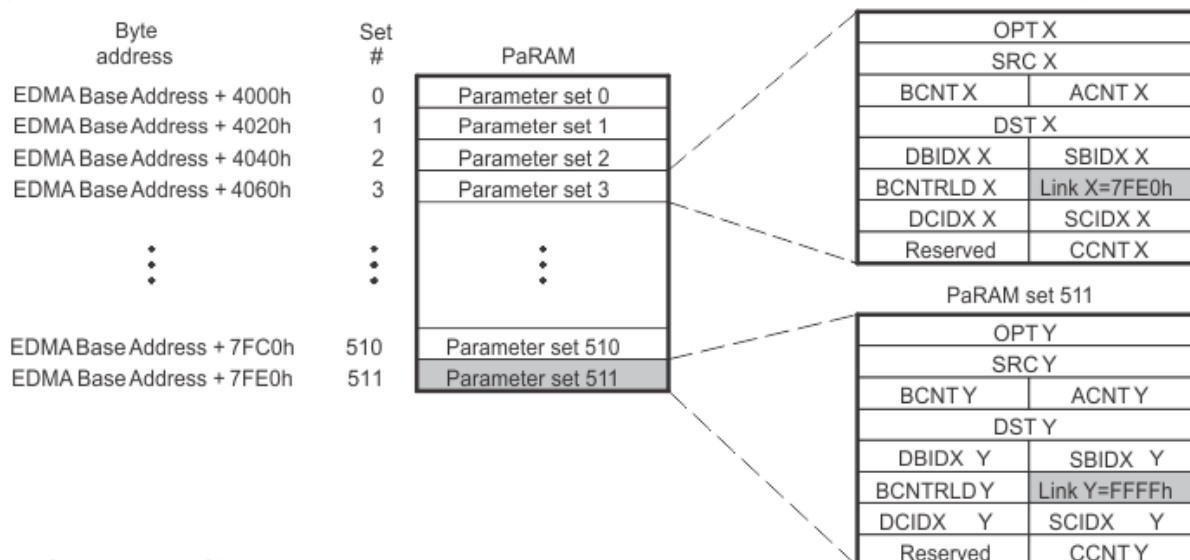
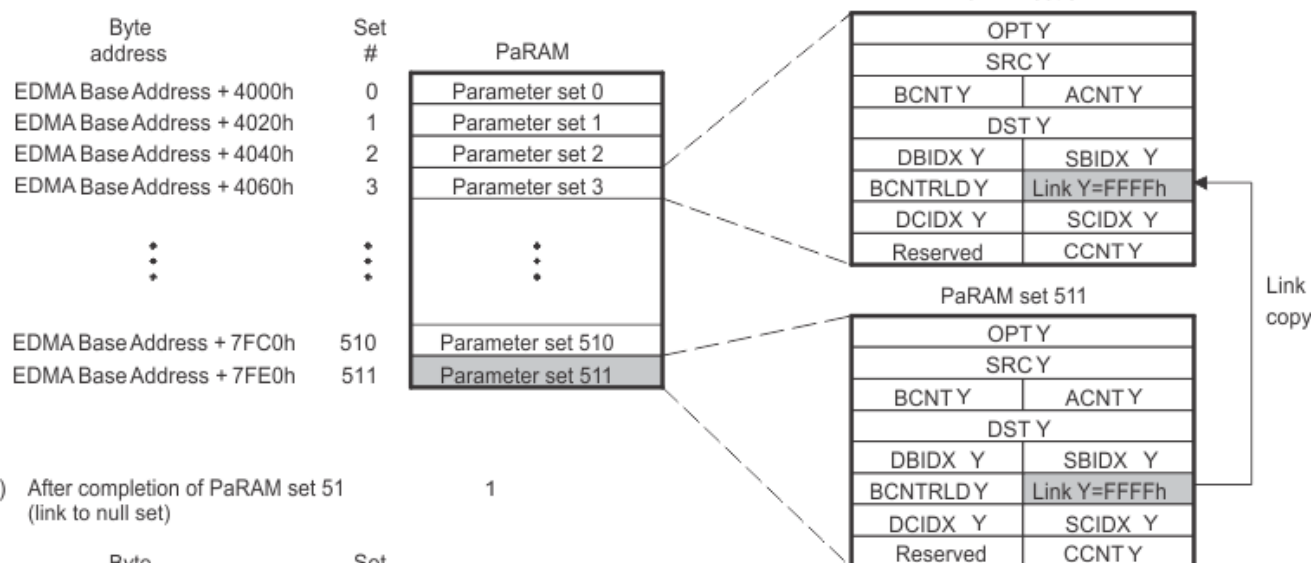
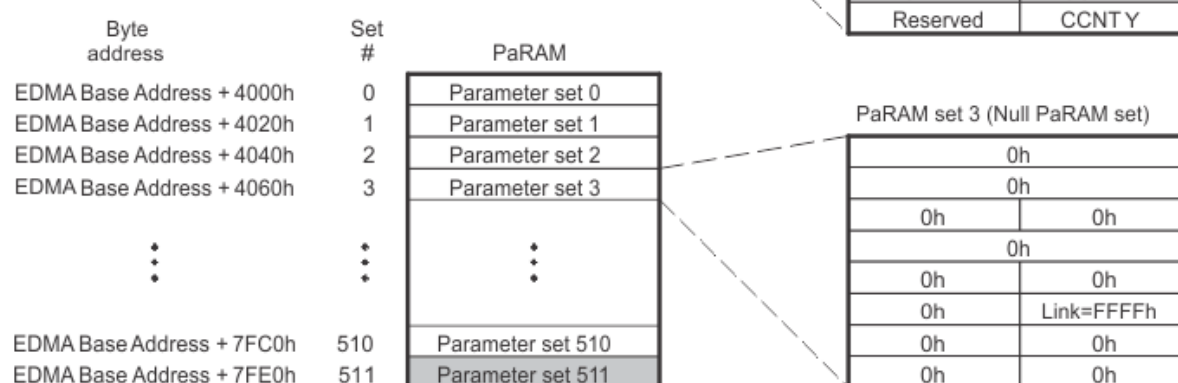
The link update occurs after the current PaRAM set event parameters have been exhausted. An event's parameters are exhausted when the EDMA channel controller has submitted all of the transfers that are associated with the PaRAM set.

A link update occurs for null and dummy transfers depending on the state of the EDMA\_TPCC\_OPT\_n[3] STATIC bit and the EDMA\_TPCC\_LNK\_n[15:0] LINK field. In both cases (null or dummy), if the value of EDMA\_TPCC\_LNK\_n[15:0] LINK is FFFFh, then a null PaRAM set (with all 0s and EDMA\_TPCC\_LNK\_n[15:0] LINK set to FFFFh) is written to the current PaRAM set.

Similarly, if EDMA\_TPCC\_LNK\_n[15:0] LINK is set to a value other than FFFFh, then the appropriate PaRAM location that EDMA\_TPCC\_LNK\_n[15:0] LINK points to is copied to the current PaRAM set.

Once the channel completion conditions are met for an event, the transfer parameters that are located at the link address are loaded into the current DMA or QDMA channel's associated parameter set. This indicates that the EDMA\_TPCC reads the entire set (eight words) from the PaRAM set specified by EDMA\_TPCC\_LNK\_n[15:0] LINK and writes all eight words to the PaRAM set that is associated with the current channel. [Figure 11-11](#) shows an example of a linked transfer.

(a) At initialization


(b) After completion of PaRAM set 3  
(link update)

(c) After completion of PaRAM set 51  
(link to null set)


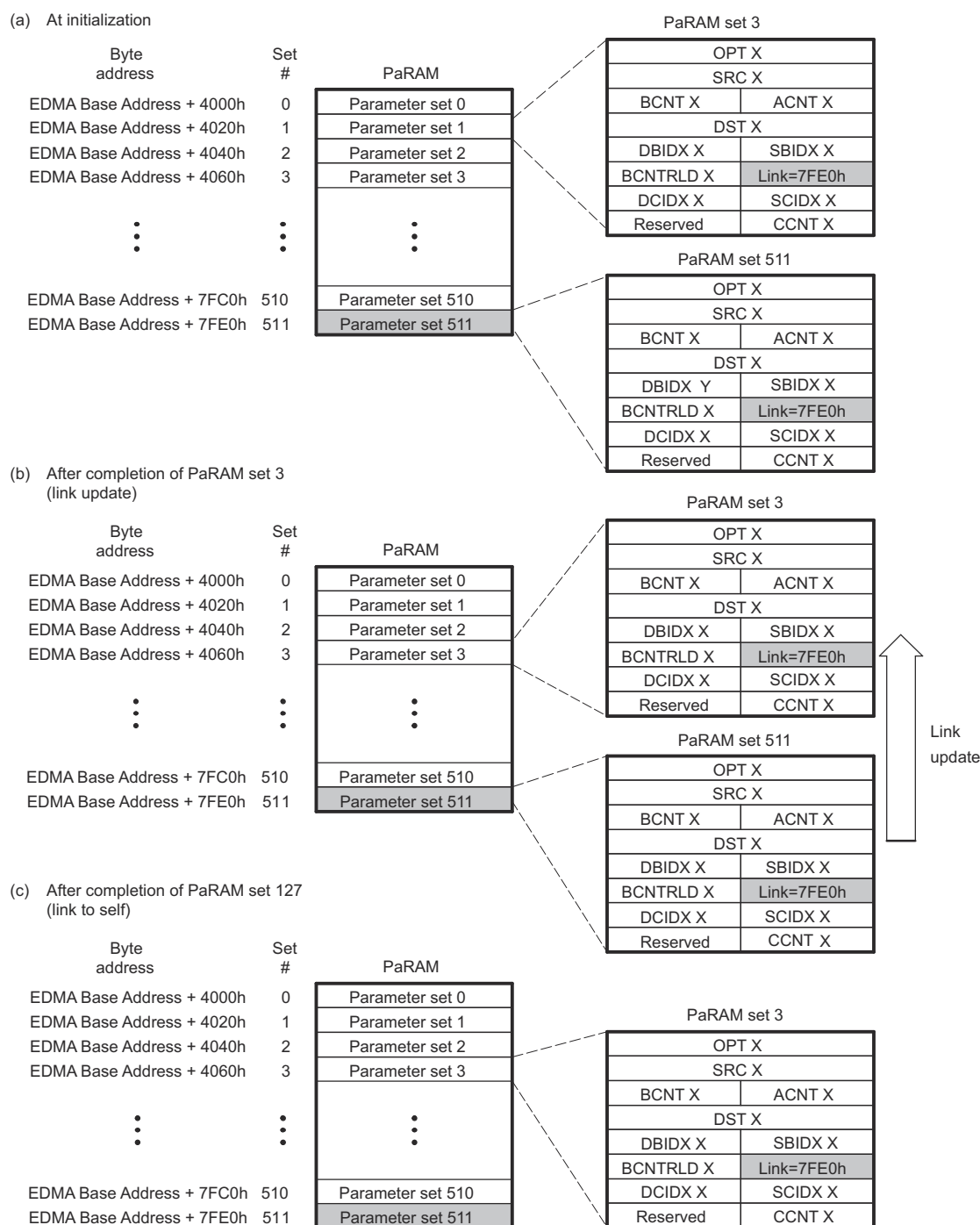
awr254-011

**Figure 11-11. Linked Transfer**

Any PaRAM set in the PaRAM can be used as a link/reload parameter set. The PaRAM sets associated with peripheral synchronization events (refer to [Section 11.3.6 Event, Channel, and PaRAM Mapping](#)) only use for linking if the corresponding events are disabled.

If a PaRAM set location is defined as a QDMA channel PaRAM set (by EDMA\_TPCC\_QCHMAPN\_j register), then copying the link PaRAM set into the current QDMA channel PaRAM set is recognized as a trigger event. It is latched in EDMA\_TPCC\_QER because a write to the trigger word was performed. This feature is used to create a linked list of transfers using a single QDMA channel and multiple PaRAM sets. Refer to [Section 11.3.4.2 QDMA Channels](#).

Linking to itself replicates the behavior of auto-initialization, thus facilitating the use of circular buffering and repetitive transfers. After an EDMA channel exhausts its current PaRAM set, it reloads all of the parameter set entries from another PaRAM set, which is initialized with values that are identical to the original PaRAM set. [Figure 11-12](#) shows an example of a linked to self transfer. Here, the PaRAM set 511 has the link field pointing to the address of parameter set 511 (linked to self).



edma-012

**Figure 11-12. Link-to-Self Transfer**

### Note

If the in EDMA\_TPCC\_OPT\_n[3] STATIC bit is set for a PaRAM set, then link updates are not performed.

### 11.3.3.8 Constant Addressing Mode Transfers/Alignment Issues

If either EDMA\_TPCC\_OPT\_n[0] SAM or EDMA\_TPCC\_OPT\_n[1] DAM is set (constant addressing mode), then the source or destination address must be aligned to a 256-bit aligned address, respectively, and the corresponding EDMA\_TPCC\_BIDX\_n is an even multiple of 32 bytes (256 bits). The EDMA\_TPCC does not recognize errors here, but the EDMA\_TPTC asserts an error if this is not true. Refer to [Section 11.3.12.3 Error Generation](#).

#### Note

The constant addressing (CONST) mode has limited applicability. The EDMA is configured for the constant addressing mode (EDMA\_TPCC\_OPT\_n[0] SAM / EDMA\_TPCC\_OPT\_n[1] DAM = 1) only if the transfer source or destination (on-chip memory, off-chip memory controllers, target peripherals) support the constant addressing mode. If the constant addressing mode is not supported, the similar logical transfer can be achieved using the increment (INCR) mode (EDMA\_TPCC\_OPT\_n[0] SAM / EDMA\_TPCC\_OPT\_n[1] DAM = 0) by appropriately programming the count and indices values.

### 11.3.3.9 Element Size

The EDMA controller does not use element-size and element-indexing. Instead, all transfers are defined in terms of all three dimensions: EDMA\_TPCC\_ABCNT\_n[15:0] ACNT, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT, and EDMA\_TPCC\_CCNT\_n[15:0] CCNT. An element-indexed transfer is logically achieved by programming EDMA\_TPCC\_ABCNT\_n[15:0] ACNT to the size of the element and EDMA\_TPCC\_ABCNT\_n[31:16] BCNT to the number of elements that need to be transferred. For example: If there are 16-bit audio data and 256 audio samples that must be transferred to a serial port, therefore the EDMA\_TPCC\_ABCNT\_n[15:0] ACNT = 2 (2 bytes) and EDMA\_TPCC\_ABCNT\_n[31:16] BCNT = 256.

### 11.3.4 Initiating a DMA Transfer

There are multiple ways to initiate a programmed data transfer using the EDMA\_TPCC channel controller. Transfers on DMA channels are initiated by three sources.

They are listed as follows:

- **Event-triggered transfer request** (this is the typical usage of EDMA controller): A peripheral, system, or externally-generated event triggers a transfer request.
- **Manually-triggered transfer request:** The CPU manually triggers a transfer by writing a 1 to the corresponding bit in the event set registers (EDMA\_TPCC\_ESR / EDMA\_TPCC\_ESRH).
- **Chain-triggered transfer request:** A transfer is triggered on the completion of another transfer or sub-transfer.

Transfers on QDMA channels are initiated by two sources. They are as follows:

- **Auto-triggered transfer request:** Writing to the programmed trigger word triggers a transfer.
- **Link-triggered transfer requests:** Writing to the trigger word triggers the transfer when linking occurs.

#### 11.3.4.1 DMA Channels

##### 11.3.4.1.1 Event-Triggered Transfer Request

When an event is asserted from a peripheral or device pins, it gets latched in the corresponding bit of the event register (EDMA\_TPCC\_ER[31:0] En = 1). For more information about peripheral events to EDMA events mapping, refer to *the device data manual*.

If the corresponding event in the event enable register (EDMA\_TPCC\_EER) is enabled (EDMA\_TPCC\_EER[31:0] En = 1), then the EDMA\_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

If the PaPARAM set is valid (not a NULL set), then a transfer request packet (TRP) is submitted to the EDMA\_TPTC and the EDMA\_TPCC\_ER[31:0] En bit is cleared. At this point, a new event can be safely received by the EDMA\_TPCC.



If the PaRAM set associated with the channel is a NULL set (see [Section 11.3.3.3 Null PaRAM Set](#)), then no transfer request (TR) is submitted and the corresponding EDMA\_TPCC\_ER[31:0] *En* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register (EDMA\_TPCC\_EMR[31:0] *En* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include cleaning the event missed error before re-triggering the DMA channel.

When an event is received, the corresponding event bit in the event register is set (EDMA\_TPCC\_ER[31:0] *En* = 1), regardless of the state of EDMA\_TPCC\_EER[31:0] *En*. If the event is disabled when an external event is received (EDMA\_TPCC\_ER[31:0] *En* = 1 and EDMA\_TPCC\_EER[31:0] *En* = 0), the EDMA\_TPCC\_ER[31:0] *En* bit remains set. If the event is subsequently enabled (EDMA\_TPCC\_EER[31:0] *En* = 1), then the pending event is processed by the EDMA\_TPCC and the TR is processed/submitted, after which the EDMA\_TPCC\_ER[31:0] *En* bit is cleared.

If an event is being processed (prioritized or is in the event queue) and another sync event is received for the same channel prior to the original being cleared (EDMA\_TPCC\_ER[31:0] *En* != 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EDMA\_TPCC\_EMR[31:0] *En* = 1).

#### 11.3.4.1.2 Manually-Triggered Transfer Request

The CPU or any peripheral device module initiates a DMA transfer by writing to the event set register EDMA\_TPCC\_ESR. Writing a 1 to an event bit in the EDMA\_TPCC\_ESR results in the event being prioritized/queued in the appropriate event queue, regardless of the state of the EDMA\_TPCC\_EER[31:0] *En* bit. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA\_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see [Section 11.3.3.3 Null PaRAM Set](#)), then no transfer request (TR) is submitted and the corresponding EDMA\_TPCC\_ER[31:0] *En* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA\_TPCC\_EMR[31:0] *En* = 1 to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include clearing the event missed error before re-triggering the DMA channel.

If an event is being processed (prioritized or is in the event queue) and the same channel is manually set by a write to the corresponding channel bit of the event set register EDMA\_TPCC\_ESR[31:0] *En* = 1 prior to the original being cleared EDMA\_TPCC\_ESR[31:0] *En* = 0, then the second event is registered as a missed event in the corresponding bit of the event missed register EDMA\_TPCC\_EMR[31:0] *En* = 1.

#### 11.3.4.1.3 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically sets the event for another channel. When a chained completion code is detected, the value of which is dictated by the transfer completion code EDMA\_TPCC\_OPT\_n[17:12] TCC of the PaRAM set associated with the channel, it results in the corresponding bit in the chained event register EDMA\_TPCC\_CER to be set EDMA\_TPCC\_CER[31:0] *E*[TCC] = 1).

Once a bit is set in EDMA\_TPCC\_CER, the EDMA\_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA\_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see [Section 11.3.3.3 Null PaRAM Set](#)), then no transfer request (TR) is submitted and the corresponding EDMA\_TPCC\_CER[31:0] *En* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA\_TPCC\_EMR[31:0] *En* = 1 to indicate that the event was discarded due to a null TR being serviced. In this case, the error condition must



be cleared before the DMA channel can be re-triggered. Good programming practices might include clearing the event missed error before re-triggering the DMA channel.

If a chaining event is being processed (prioritized or queued) and another chained event is received for the same channel prior to the original being cleared  $EDMA\_TPCC\_CER[31:0] \text{ } En \neq 0$ , then the second chained event is registered as a missed event in the corresponding channel bit of the event missed register  $EDMA\_TPCC\_EMR[31:0] \text{ } En = 1$ .

---

#### Note

Chained event registers  $EDMA\_TPCC\_CER$ , event registers  $EDMA\_TPCC\_ER$ , and event set registers  $EDMA\_TPCC\_ESR$  operate independently. An event  $En$  can be triggered by any of the trigger sources (event-triggered, manually-triggered, or chain-triggered).

---

### 11.3.4.2 QDMA Channels

#### 11.3.4.2.1 Auto-Triggered and Link-Triggered Transfer Request

QDMA-based transfer requests are issued when a QDMA event gets latched in the QDMA event register  $EDMA\_TPCC\_QER[31:0] \text{ } En = 1$ . A bit corresponding to a QDMA channel is set in the QDMA event register  $EDMA\_TPCC\_QER$  when the following occurs:

- A CPU (or any device module) write occurs to a PaRAM address that is defined as a QDMA channel trigger word (programmed in the QDMA channel mapping register  $EDMA\_TPCC\_QCHMAPN\_j$  for the particular QDMA channel and the QDMA channel is enabled via the QDMA event enable register  $EDMA\_TPCC\_QEER[31:0] \text{ } En = 1$ .
- $EDMA\_TPCC$  performs a link update on a PaRAM set address that is configured as a QDMA channel matches  $EDMA\_TPCC\_QCHMAPN\_j$  settings and the corresponding channel is enabled via the QDMA event enable register  $EDMA\_TPCC\_QEER[31:0] \text{ } En = 1$ .

Once a bit is set in  $EDMA\_TPCC\_QER$ , the  $EDMA\_TPCC$  prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated  $EDMA\_TPTC$  and the channel can be triggered again.

If a bit is already set in  $EDMA\_TPCC\_QER[31:0] \text{ } En = 1$  and a second QDMA event for the same QDMA channel occurs prior to the original being cleared, the second QDMA event gets captured in the QDMA event miss register  $EDMA\_TPCC\_QEMR[7:0] \text{ } En = 1$ .

#### 11.3.4.3 Comparison Between DMA and QDMA Channels

The primary difference between DMA and QDMA channels is the event/channel synchronization.

QDMA events are either auto-triggered or link triggered. Auto-triggering allows QDMA channels to be triggered by CPU(s) with a minimum number of linear writes to PaRAM. Link triggering allows a linked list of transfers to be executed, using a single QDMA PaRAM set and multiple link PaRAM sets.

A QDMA transfer is triggered when a CPU (or other device modules) writes to the trigger word of the QDMA channel parameter set (auto-triggered) or when the  $EDMA\_TPCC$  performs a link update on a PaRAM set that has been mapped to a QDMA channel (link triggered).

---

#### Note

The CPUs triggered (manually triggered) DMA channels, in addition to writing to the PaRAM set, it is required to write to the event set register  $EDMA\_TPCC\_ESR$  to kick-off the transfer.

---

QDMA channels are typically for cases where a single event accomplishes a complete transfer since the CPU (or other device modules) must reprogram some portion of the QDMA PaRAM set in order to re-trigger the channel. QDMA transfers are programmed with  $EDMA\_TPCC\_ABCNT\_n[31:0] \text{ } BCNT = 1$  and

EDMA\_TPCC\_CCNT\_n[15:0] CCNT = 1 for A-synchronized transfers, and EDMA\_TPCC\_CCNT\_n[15:0] CCNT = 1 for AB-synchronized transfers.

Additionally, since linking is also supported (if EDMA\_TPCC\_OPT\_n[3] STATIC = 0) for QDMA transfers, it allows to initiate a linked list of QDMAs, so when EDMA\_TPCC copies over a link PaRAM set (including the write to the trigger word), the current PaRAM set mapped to the QDMA channel automatically recognizes as a valid QDMA event and initiate another set of transfers as specified by the linked set.

### 11.3.5 Completion of a DMA Transfer

A parameter set for a given channel is complete when the required number of transfer requests is submitted (based on receiving the number of synchronization events). The expected number of TRs for a non-null/non-dummy transfer is shown in Table 11-15 for both synchronization types along with state of the PaRAM set prior to the final TR being submitted. When the counts (EDMA\_TPCC\_ABCNT\_n[31:0] BCNT and/or EDMA\_TPCC\_CCNT\_n[15:0] CCNT) are this value, the next TR results in:

- Final chaining or interrupt codes sent by the transfer controllers (instead of intermediate).
- Link updates (linking to either null or another valid link set).

**Table 11-15. Expected Number of Transfers for Non-Null Transfer**

Sync Mode	Counts at time 0	Total # Transfers	Counts prior to final TR
A-synchronized	ACNT BCNT CCNT	(BCNT × CCNT ) TRs of ACNT bytes each	EDMA_TPCC_ABCNT_n[31:0] BCNT == 1 && EDMA_TPCC_CCNT_n[15:0] CCNT == 1
AB-synchronized	ACNT BCNT CCNT	CCNT TRs for ACNT × BCNT bytes each	EDMA_TPCC_CCNT_n[15:0] CCNT == 1

The PaRAM OPT field must program with a specific transfer completion code TCC or EDMA\_TPCC\_OPT\_n[17:12] TCC along with the other EDMA\_TPCC\_OPT\_n fields ([22] TCCHEN, [20] TCINTEN, [23] ITCCHEN, and [21] ITCINTEN bits) to indicate whether the completion code is to be used for generating a chained event or/and for generating an interrupt upon completion of a transfer.

The specific EDMA\_TPCC\_OPT\_n[17:12] TCC value (6-bit binary value) programmed dictates which of the 64-bits in the chain event register EDMA\_TPCC\_CER [TCC] and/or interrupt pending register EDMA\_TPCC\_IPR [TCC] is set.

It can selectively program whether the transfer controller sends back completion codes on completion of the final transfer request (TR) of a parameter set EDMA\_TPCC\_OPT\_n[22] TCCHEN or EDMA\_TPCC\_OPT\_n[20] TCINTEN, for all but the final transfer request (TR) of a parameter set EDMA\_TPCC\_OPT\_n[23] ITCCHEN or EDMA\_TPCC\_OPT\_n[21] ITCINTEN, or for all TRs of a parameter set (both). Refer to Section 11.3.8 Chaining EDMA Channels for details on chaining (intermediate/final chaining) and Section 11.3.9 EDMA Interrupts for details on intermediate/final interrupt completion.

A completion detection interface exists between the EDMA channel controller and transfer controller(s). This interface sends back information from the transfer controller to the channel controller to indicate that a specific transfer is completed. Completion of a transfer is used for generating chained events and/or generating interrupts to the CPU(s).

All DMA/QDMA PaRAM sets must also specify a link address value. For repetitive transfers such as ping-pong buffers, the link address value must point to another predefined PaRAM set. Alternatively, a non-repetitive transfer must set the link address value to the null link value. The null link value is defined as FFFFh. Refer to Section 11.3.3.7 Linking Transfers for more details.

### Note

Any incoming events that are mapped to a null PaRAM set results in an error condition. The error condition must clear before the corresponding channel is used again. Refer to [Section 11.3.3.5 Dummy Versus Null Transfer Comparison](#).

There are three ways the EDMA\_TPCC gets updated/informed about a transfer completion: normal completion, early completion, and dummy/null completion. This applies to both chained events and completion interrupt generation.

#### 11.3.5.1 Normal Completion

In normal completion mode EDMA\_TPCC\_OPT\_n[11] TCCMODE = 0, the transfer or sub-transfer is considered to be complete when the EDMA channel controller receives the completion codes from the EDMA transfer controller. In this mode, the completion code to the channel controller is posted by the transfer controller after it receives a signal from the destination peripheral. Normal completion is typically used to generate an interrupt to inform the CPU that a set of data is ready for processing.

#### 11.3.5.2 Early Completion

In early completion mode EDMA\_TPCC\_OPT\_n[11] TCCMODE = 1, the transfer is considered to be complete when the EDMA channel controller submits the transfer request (TR) to the EDMA transfer controller. In this mode, the channel controller generates the completion code internally. Early completion is typically useful for chaining, as it allows subsequent transfers to be chained-triggered while the previous transfer is still in progress within the transfer controller, maximizing the overall throughput of the set of the transfers.

#### 11.3.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set [Section 11.3.3.4](#) or null set [Section 11.3.3.3](#). In both cases, the EDMA channel controller does not submit the associated transfer request to the EDMA transfer controller(s). However, if the set (dummy/null) has the OPT field programmed to return completion code (intermediate/final interrupt/chaining completion), then it sets the appropriate bits in the interrupt pending registers EDMA\_TPCC\_IPR and EDMA\_TPCC\_IPRH or chained event register EDMA\_TPCC\_CER and EDMA\_TPCC\_CERH. The internal early completion path is used by the channel controller to return the completion codes internally (that is, EDMA\_TPCC generates the completion code).

### 11.3.6 Event, Channel, and PaRAM Mapping

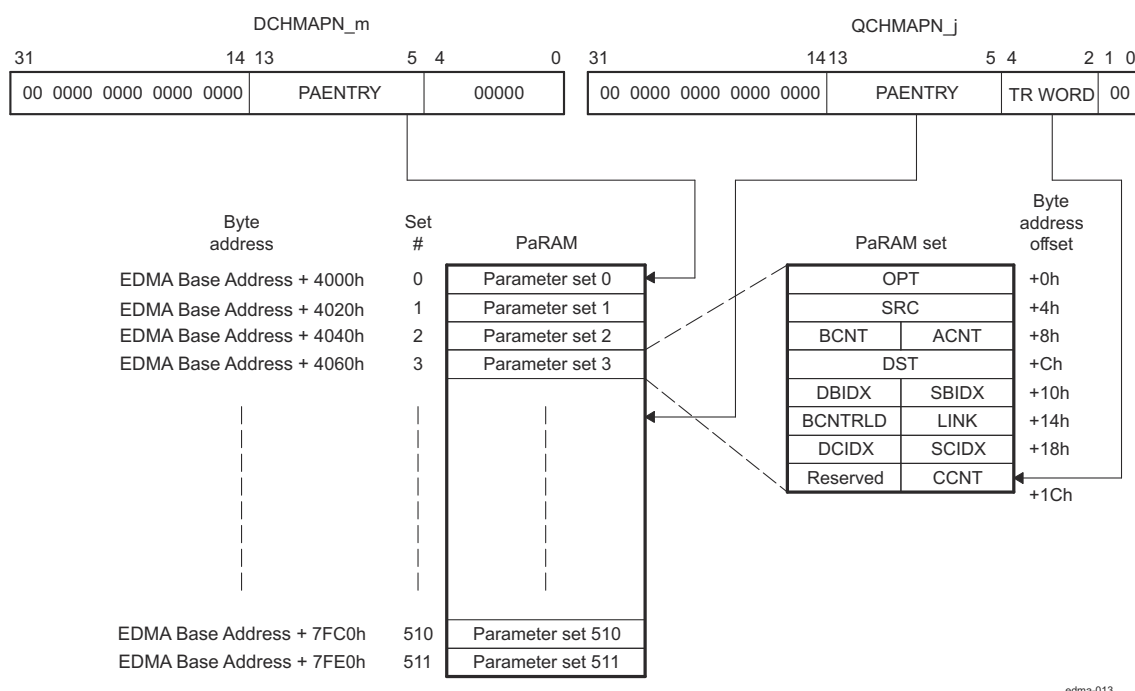
Several of the 64 DMA channels are tied to a specific hardware event, thus allowing events from device peripherals or external hardware (via the dma\_evt[3:0] pins) to trigger transfers. A DMA channel typically requests a data transfer when it receives its event (apart from manually-triggered, chain-triggered, and other transfers). The amount of data transferred per synchronization event depends on the channel's configuration (EDMA\_TPCC\_ABCNT\_n[15:0] ACNT, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT, EDMA\_TPCC\_CCNT\_n[15:0] CCNT, etc.) and the synchronization type (A-synchronized or AB-synchronized).

The association of an event to a channel is fixed within the EDMA Channel Controller, that is, each DMA channel has one specific event associated with it.

In an application, if a channel does not use the associated synchronization event or if it does not have an associated synchronization event (unused), that channel can be used for manually-triggered or chained-triggered transfers, for linking/reloading, or as a QDMA channel.

#### 11.3.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see [Section 11.3.3](#)). The DMA channel mapping registers EDMA\_TPCC\_DCHMAPN\_m in the EDMA\_TPCC provide programmability that allows the DMA channels to be mapped to any of the PaRAM sets in the PaRAM memory map. [Figure 11-13](#) illustrates the use of EDMA\_TPCC\_DCHMAPN\_m. There is one EDMA\_TPCC\_DCHMAPN\_m register per channel.


**Figure 11-13. DMA Channel and QDMA Channel to PaRAM Mapping**

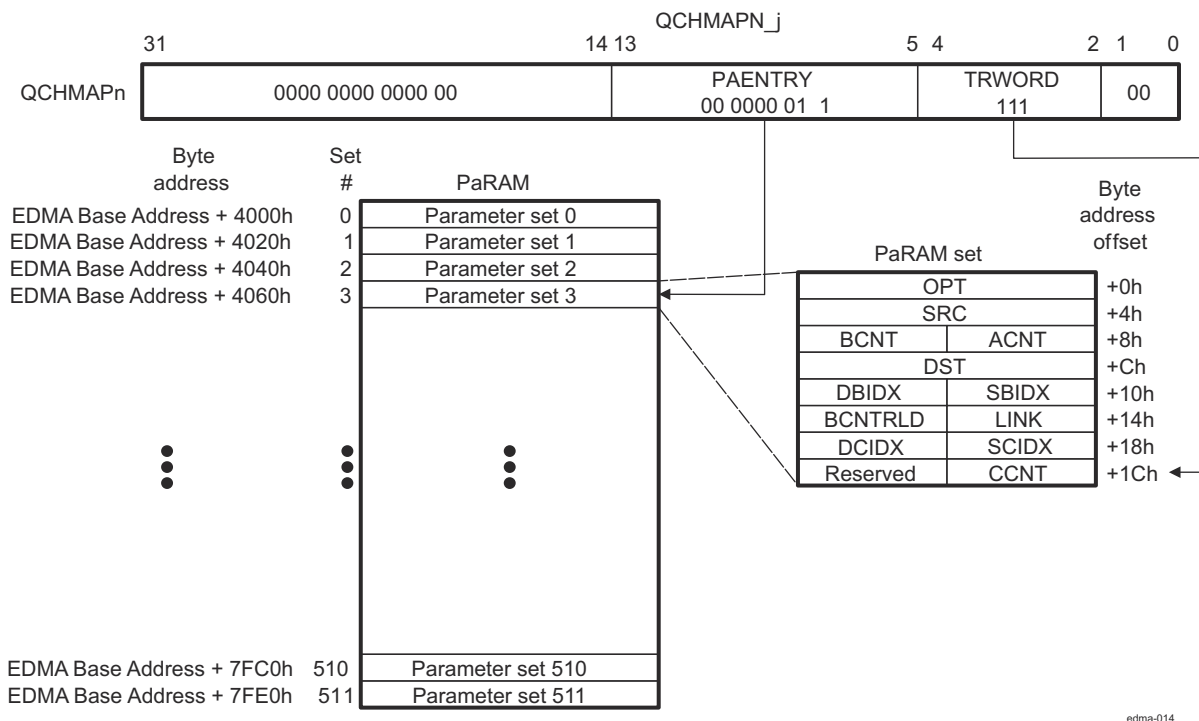
### Note

#### 11.3.6.2 QDMA Channel to PaRAM Mapping

The mapping between the QDMA channels and the PaRAM sets is programmable. The QDMA channel mapping register `EDMA_TPCC_QCHMAPN_j` in the `EDMA_TPCC` allows to map the QDMA channels to any of the PaRAM sets in the PaRAM memory map. [Figure 11-14](#) illustrates the use of `EDMA_TPCC_QCHMAPN_j`.

`EDMA_TPCC_QCHMAPN_j[4:2]` TRWORD bit-field allows to program the trigger word in the PaRAM set for the QDMA channel. A trigger word is one of the eight words in the PaRAM set. For a QDMA transfer to occur, a valid TR synchronization event for `EDMA_TPCC` is a write to the trigger word in the PaRAM set pointed to by `EDMA_TPCC_QCHMAPN_j` for a particular QDMA channel. By default, QDMA channels are mapped to PaRAM set 0.

It must appropriately re-map PaRAM set 0 before use.



**Figure 11-14. QDMA Channel to PaRAM Mapping**

### 11.3.7 EDMA Channel Controller Regions

The EDMA channel controller divides its address space into eight regions. Individual channel resources are assigned to a specific region, where each region is typically assigned to a specific device module uses the EDMA controller.

Application software can use regions or to ignore them altogether. It can be used active memory protection in conjunction with regions so that only a specific device module which uses the EDMA (for example, privilege identification) or privilege level (for example, user vs. supervisor) is allowed access to a given region, and thus to a given DMA or QDMA channel. This allows robust system-level DMA code where each EDMA initiator only modifies the state of the assigned resources. Memory protection is described in [Section 11.3.10 Memory Protection](#).

#### 11.3.7.1 Region Overview

The EDMA channel controller memory-mapped registers are divided in three main categories:

1. Global registers
2. Global region channel registers
3. Shadow region channel registers

The global registers are located at a single/fixed location in the EDMA\_TPCC memory map. These registers control EDMA resource mapping and provide debug visibility and error tracking information.

The channel registers (including DMA, QDMA, and interrupt registers) are accessible via the global channel region address range, or in the shadow *n* channel region address range(s). For example, the event enable register EDMA\_TPCC\_EER is visible at the global address of EDMA Base Address + 1020h or region addresses of EDMA Base Address + 2020h for region 0, EDMA Base Address + 2220h for region 1, ... EDMA Base Address + 2E20h for region 7.

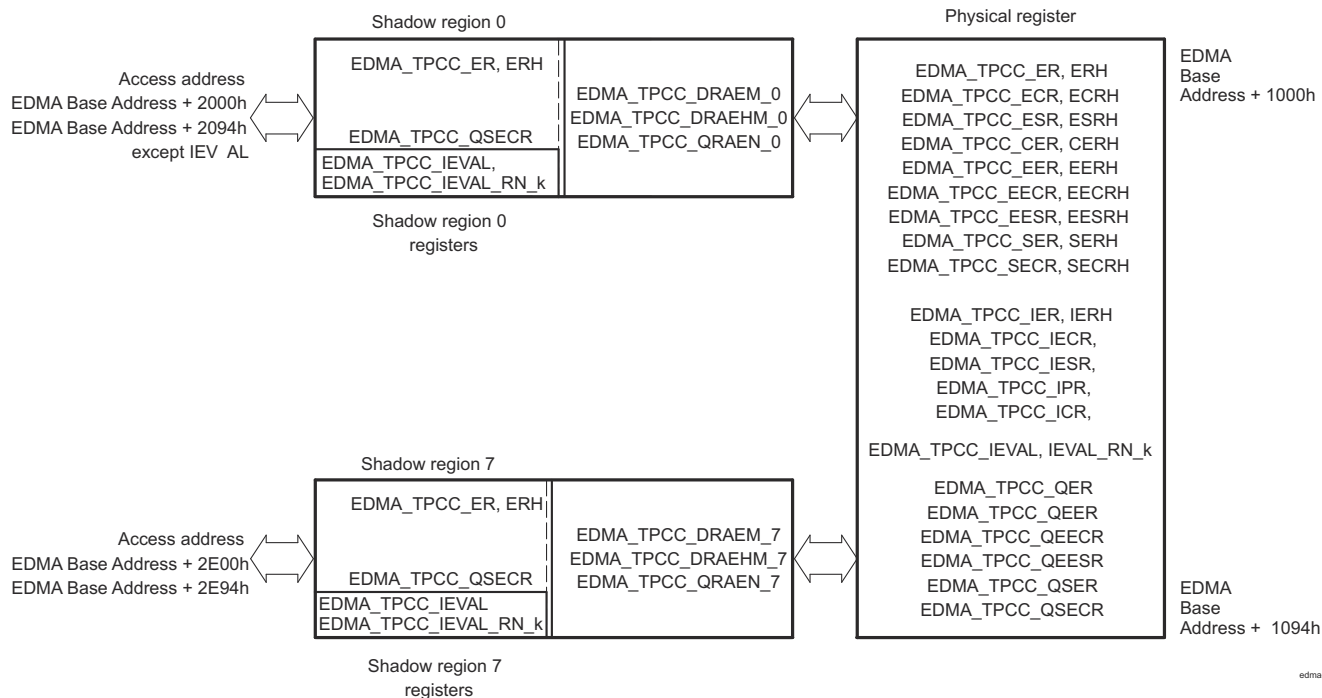
The DMA region access enable registers EDMA\_TPCC\_DRAEM<sub>k</sub> and the QDMA region access enable registers EDMA\_TPCC\_QRAEN<sub>k</sub> control the underlying control register bits that are accessible via the shadow region address space (except for EDMA\_TPCC\_IEVAL and EDMA\_TPCC\_IEVAL\_RN<sub>k</sub> registers). [Table 11-16](#)

lists the registers in the shadow region memory map. Refer to *EDMA\_TPCC register mapping summary* for the complete global and shadow region memory maps.

**Table 11-16. Shadow Region Registers**

EDMA_TPCC_DRAE M_k	EDMA_TPCC_DRAE HM_k	EDMA_TPCC_QRAE N_k
EDMA_TPCC_ER	EDMA_TPCC_ERH	EDMA_TPCC_QER
EDMA_TPCC_ECR	EDMA_TPCC_ECRH	EDMA_TPCC_QEER
EDMA_TPCC_ESR	EDMA_TPCC_ESRH	EDMA_TPCC_QEEC R
EDMA_TPCC_CER	EDMA_TPCC_CERH	EDMA_TPCC_QEES R
EDMA_TPCC_EER	EDMA_TPCC_EERH	
EDMA_TPCC_EECR	EDMA_TPCC_EECR H	
EDMA_TPCC_EESR	EDMA_TPCC_EESR H	
EDMA_TPCC_SER	EDMA_TPCC_SERH	
EDMA_TPCC_SECR	EDMA_TPCC_SECR H	
EDMA_TPCC_IER	EDMA_TPCC_IERH	
EDMA_TPCC_IECR	EDMA_TPCC_IECRH	
EDMA_TPCC_IESR	EDMA_TPCC_IESRH	
EDMA_TPCC_IPR	EDMA_TPCC_IPRH	
EDMA_TPCC_ICR	EDMA_TPCC_ICRH	
<b>Register not affected by DRAE/DRAEH</b>		
EDMA_TPCC_IEVAL		
EDMA_TPCC_IEVAL _RN_k		

Figure 11-15 illustrates the conceptual view of the regions.



**Figure 11-15. Shadow Region Registers**

### 11.3.7.2 Channel Controller Regions

There are eight EDMA shadow regions (and associated memory maps). Associated with each shadow region are a set of registers defining which channels and interrupt completion codes belong to that region. These registers are user-programmed per region to assign ownership of the DMA/QDMA channels to a region.

- **EDMA\_TPCC\_DRAEM\_k and EDMA\_TPCC\_DRAEHM\_k:** One register pair exists for each of the shadow regions. The number of bits in each register pair matches the number of DMA channels (64 DMA channels). These registers need to be programmed to assign ownership of DMA channels and interrupt (or EDMA\_TPCC\_OPT\_n[17:12] TCC codes) to the respective region. Accesses to DMA and interrupt registers via the shadow region address view are filtered through the DRAEM/DRAEHM pair. A value of 1 in the corresponding EDMA\_TPCC\_DRAEM\_k[31:0] / EDMA\_TPCC\_DRAEHM\_k[31:0] bit implies that the corresponding DMA interrupt channel is accessible; a value of 0 in the corresponding EDMA\_TPCC\_DRAEM\_k[31:0] / EDMA\_TPCC\_DRAEHM\_k[31:0] bit forces writes to be discarded and returns a value of 0 for reads.
- **EDMA\_TPCC\_QRAEN\_k:** One register exists for every region. The number of bits in each register matches the number of QDMA channels (8 QDMA channels). These registers must be programmed to assign ownership of QDMA channels to the respective region. To enable a channel in a shadow region using shadow region 0 EDMA\_TPCC\_QEER, the corresponding bits in QRAE must be set or writing into EDMA\_TPCC\_QEESR there will be no the desired effect.
- **EDMA\_TPCC\_MPPAN\_k and EDMA\_TPCC\_MPPAG:** One register exists for every region. This register defines the privilege level, requestor, and types of accesses allowed to a region's memory-mapped registers.

It is typical for an application to have a unique assignment of QDMA/DMA channels (and, therefore, a given bit position) to a given region.

The use of shadow regions allows restricted access to EDMA resources (DMA channels, QDMA channels, TCC, interrupts) by tasks in a system by setting or clearing bits in the EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_QRAEN\_k registers.

If exclusive access to any given channel / TCC code is required for a region, then only that region's EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_QRAEN\_k have the associated bit set.



### Example 11-1. Resource Pool Division Across Two Regions

This example illustrates a resource pool division across two regions, assuming region 0 must be allocated 16 DMA channels (0-15) and 1 QDMA channel (0) and 32 TCC codes (0-15 and 48-63).

Region 1 needs to be allocated 16 DMA channels (16-32) and the remaining 7 QDMA channels (1-7) and TCC codes (16-47).

EDMA\_TPCC\_DRAEM\_k should be equal to the OR of the bits that are required for the DMA channels and the TCC codes:

Region 0: DRAEHM, DRAEM = 0xFFFF0000, 0x0000FFFF QRAEN = 0x00000001 Region 1: DRAEHM, DRAEM = 0x0000FFFF, 0xFFFF0000 QRAEN = 0x000000FE
--

#### 11.3.7.3 Region Interrupts

In addition to the EDMA\_TPCC global completion interrupt, there is an additional completion interrupt line that is associated with every shadow region. Along with the interrupt enable register EDMA\_TPCC\_IER, DRAEM acts as a secondary interrupt enable for the respective shadow region interrupts. Refer to *Hardware Request* for more information about EDMA Interrupts.

#### 11.3.8 Chaining EDMA Channels

The channel chaining capability for the EDMA allows the completion of an EDMA channel transfer to trigger another EDMA channel transfer. The purpose is to allow the ability to chain several events through one event occurrence.

Chaining is different from linking ([Section 11.3.3.7 Linking Transfers](#)). The EDMA link feature reloads the current channel parameter set with the linked parameter set. The EDMA chaining feature does not modify or update any channel parameter set. It provides a synchronization event to the chained channel (see [Section 11.3.4.1.3 Chain-Triggered Transfer Request](#)).

Chaining is achieved at either final transfer completion or intermediate transfer completion, or both, of the current channel. Consider a channel  $m$  (DMA/QDMA) required to chain to channel  $n$ . Channel number  $n$  (0-63) needs to be programmed into the EDMA\_TPCC\_OPT\_n[17:12] TCC bit-field of channel  $m$  channel options parameter (OPT) set.

- If final transfer completion chaining EDMA\_TPCC\_OPT\_n[22] TCCHEN = 1 is enabled, the chain-triggered event occurs after the submission of the last transfer request of channel  $m$  is either submitted or completed (depending on early or normal completion).



- If intermediate transfer completion chaining EDMA\_TPCC\_OPT\_n[23] ITCCHEN = 1 is enabled, the chain-triggered event occurs after every transfer request, except the last of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion chaining (EDMA\_TPCC\_OPT\_n[22] TCCHEN = 1 and EDMA\_TPCC\_OPT\_n[23] ITCCHEN = 1) are enabled, then the chain-trigger event occurs after every transfer request is submitted or completed (depending on early or normal completion).

**Table 11-17** illustrates the number of chain event triggers occurring in different synchronized scenarios. Consider channel 31 programmed with EDMA\_TPCC\_ABCNT\_n[15:0] ACNT = 3, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT = 4, EDMA\_TPCC\_CCNT\_n[15:0] CCNT = 5, and EDMA\_TPCC\_OPT\_n[17:12] TCC = 30.

**Table 11-17. Chain Event Triggers**

Options	(Number of chained event triggers on channel 30)	
	A-Synchronized	AB-Synchronized
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 0	1 (Owing to the last TR)	1 (Owing to the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 0, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	19 (Owing to all but the last TR)	4 (Owing to all but the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	20 (Owing to a total of 20 TRs)	5 (Owing to a total of 5 TRs)

### 11.3.9 EDMA Interrupts

The EDMA interrupts are divided into 2 categories: transfer completion interrupts and error interrupts.

There are nine region interrupts, eight shadow regions and one global region. The transfer completion interrupts are listed in **Table 11-18**. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the device interrupt controllers INTCs.

**Table 11-18. EDMA Transfer Completion Interrupts**

Name	Description
EDMA_TPCC_INT0	EDMA_TPCC Transfer Completion Interrupt Shadow Region 0
EDMA_TPCC_INT1	EDMA_TPCC Transfer Completion Interrupt Shadow Region 1
EDMA_TPCC_INT2	EDMA_TPCC Transfer Completion Interrupt Shadow Region 2
EDMA_TPCC_INT3	EDMA_TPCC Transfer Completion Interrupt Shadow Region 3
EDMA_TPCC_INT4	EDMA_TPCC Transfer Completion Interrupt Shadow Region 4
EDMA_TPCC_INT5	EDMA_TPCC Transfer Completion Interrupt Shadow Region 5
EDMA_TPCC_INT6	EDMA_TPCC Transfer Completion Interrupt Shadow Region 6
EDMA_TPCC_INT7	EDMA_TPCC Transfer Completion Interrupt Shadow Region 7

**Table 11-19. EDMA Error Interrupts**

Name	Description
EDMA_TPCC_ERRINT	EDMA_TPCC Error Interrupt
EDMA_TPCC_MPINT	EDMA_TPCC Memory Protection Interrupt
EDMA_TC0_ERRINT	TC0 Error Interrupt
EDMA_TC1_ERRINT	TC1 Error Interrupt

#### 11.3.9.1 Transfer Completion Interrupts

The EDMA\_TPCC is responsible for generating transfer completion interrupts to the CPU(s) (and other EDMA controllers). The EDMA generates a single completion interrupt per shadow region, as well as one for the global region on behalf of all 64 channels. The various control registers and bit fields facilitate EDMA interrupt generation.

The software architecture must either use the global interrupt or the shadow interrupts, but not both.

The transfer completion code EDMA\_TPCC\_OPT\_n[17:12] TCC value is directly mapped to the bits of the interrupt pending register EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH.

For example, if EDMA\_TPCC\_OPT\_n[17:12] TCC = 10 0001b, EDMA\_TPCC\_IPRH[1] is set after transfer completion, and results in interrupt generation to the CPU(s) if the completion interrupt is enabled for the CPU. See [Section 11.3.9.1.1 Enabling Transfer Completion Interrupts](#) for details about enabling EDMA transfer completion interrupts.

When a completion code is returned (as a result of early or normal completions), the corresponding bit in EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH registers is set if transfer completion interrupt (final/intermediate) is enabled in the channel options parameter (OPT) for a PaRAM set associated with the transfer.

**Table 11-20. Transfer Complete Code (TCC) to EDMA\_TPCC Interrupt Mapping**

TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPR Bit Set	TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPRH Bit Set <sup>(1)</sup>
0	EDMA_TPCC_IPR[0]	20h	EDMA_TPCC_IPR[32] / EDMA_TPCC_IPRH[0]
1	EDMA_TPCC_IPR[1]	21h	EDMA_TPCC_IPR[33] / EDMA_TPCC_IPRH[1]
2h	EDMA_TPCC_IPR[2]	22h	EDMA_TPCC_IPR[34] / EDMA_TPCC_IPRH[2]
3h	EDMA_TPCC_IPR[3]	23h	EDMA_TPCC_IPR[35] / EDMA_TPCC_IPRH[3]
4h	EDMA_TPCC_IPR[4]	24h	EDMA_TPCC_IPR[36] / EDMA_TPCC_IPRH[4]
...	...	...	...
1Eh	EDMA_TPCC_IPR[30]	3Eh	EDMA_TPCC_IPR[62] / EDMA_TPCC_IPRH[30]
1Fh	EDMA_TPCC_IPR[31]	3Fh	EDMA_TPCC_IPR[63] / EDMA_TPCC_IPRH[31]

(1) Bit fields EDMA\_TPCC\_IPR [32-63] correspond to bits 0 to 31 in EDMA\_TPCC\_IPRH, respectively.

The transfer completion code (TCC) can program to any value for a DMA/QDMA channel. A direct relation between the channel number and the transfer completion code value does not need to exist. This allows multiple channels having the same transfer completion code value to cause a CPU to execute the same interrupt service routine (ISR) for different channels.

If the channel is used in the context of a shadow region and it intends for the shadow region interrupt to be asserted, then ensure that the bit corresponding to the TCC code is enabled in EDMA\_TPCC\_IER / EDMA\_TPCC\_IERH and in the corresponding shadow region's DMA region access registers (EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k).

Interrupt generation can be enabled at either final transfer completion or intermediate transfer completion, or both. Consider channel *m* as an example.

- If the final transfer interrupt (EDMA\_TPCC\_OPT\_n[20] TCINTEN = 1) is enabled, the interrupt occurs after the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).
- If the intermediate transfer interrupt (EDMA\_TPCC\_OPT\_n[21] ITCINTEN = 1) is enabled, the interrupt occurs after every transfer request, except the last TR of channel *m* is either submitted or completed (depending on early or normal completion).

- If both final and intermediate transfer completion interrupts (EDMA\_TPCC\_OPT\_n[20] TCINTEN = 1, and EDMA\_TPCC\_OPT\_n[21] ITCINTEN = 1) are enabled, then the interrupt occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 11-21 shows the number of interrupts that occur in different synchronized scenarios. Consider channel 31, programmed with ABCNT\_n[15:0] ACNT = 3, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT = 4, EDMA\_TPCC\_CCNT\_n[15:0]CCNT = 5, and EDMA\_TPCC\_OPT\_n[17:12] TCC = 30.

**Table 11-21. Number of Interrupts**

Options	A-Synchronized	AB-Synchronized
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 0	1 (Last TR)	1 (Last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 0, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	19 (All but the last TR)	4 (All but the last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	20 (All TRs)	5 (All TRs)

#### 11.3.9.1.1 Enabling Transfer Completion Interrupts

For the EDMA channel controller to assert a transfer completion to the external environment, the interrupts must be enabled in the EDMA\_TPCC. This is in addition to setting up the EDMA\_TPCC\_OPT\_n[20] TCINTEN and EDMA\_TPCC\_OPT\_n[21] ITCINTEN bits of the associated PaRAM set.

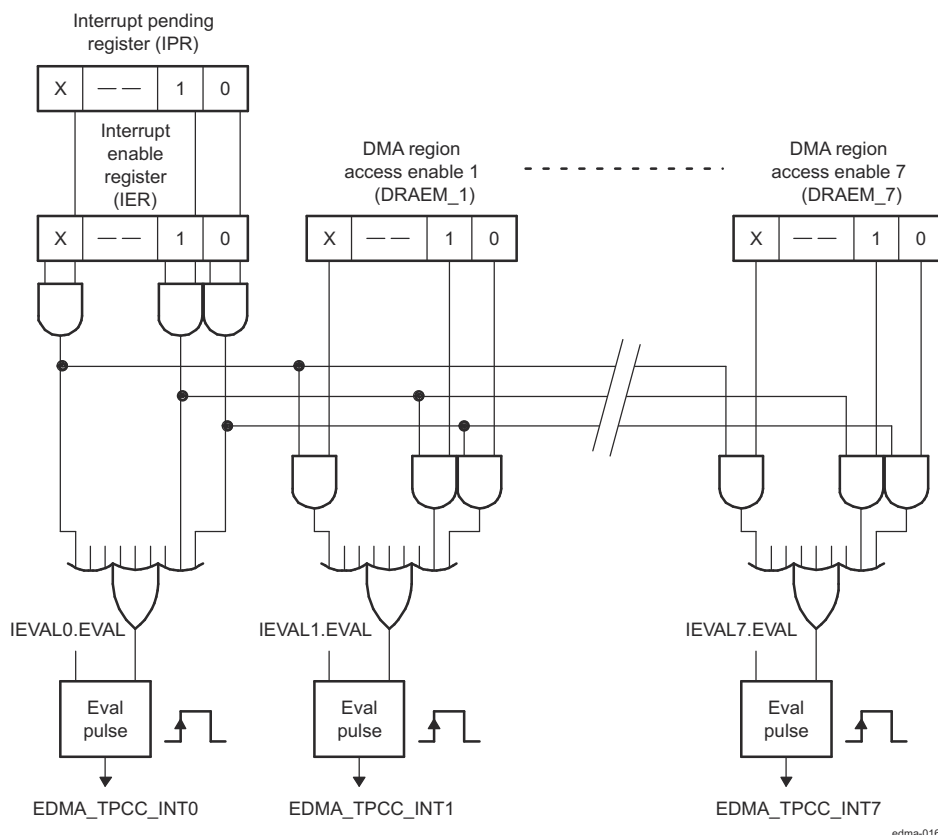
The EDMA channel controller has interrupt enable registers EDMA\_TPCC\_IER / EDMA\_TPCC\_IERH and each bit location in EDMA\_TPCC\_IER / EDMA\_TPCC\_IERH serves as a primary enable for the corresponding interrupt pending registers EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH.

All of the interrupt registers (EDMA\_TPCC\_IER, EDMA\_TPCC\_IESR, EDMA\_TPCC\_IECR, and EDMA\_TPCC\_IPR) are either manipulated from the global DMA channel region, or by the DMA channel shadow regions. The shadow regions provide a view to the same set of physical registers that are in the global region.

The EDMA channel controller has a hierarchical completion interrupt scheme that uses a single set of interrupt pending registers EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH and single set of interrupt enable registers EDMA\_TPCC\_IER / EDMA\_TPCC\_IERH. The programmable DMA region access enable registers EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k provides a second level of interrupt masking. The global region interrupt output is gated based on the enable mask that is provided by EDMA\_TPCC\_IER / EDMA\_TPCC\_IERH, see [Figure 11-16](#)

The region interrupt outputs are gated by EDMA\_TPCC\_IER and the specific EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k associated with the region.

[Figure 11-16](#) shows the Interrupt diagram of the EDMA controller.



**Figure 11-16. Interrupt Diagram**

The EDMA\_TPCC generates the transfer completion interrupts that are associated with each shadow region, the following conditions must be true:

- EDMA\_TPCC\_INT0: (EDMA\_TPCC\_IPR[0] E0 & EDMA\_TPCC\_IER[0] E0 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_0[0] E0) | (EDMA\_TPCC\_IPR[1] E1 & EDMA\_TPCC\_IER[1] E1 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_0[1] E1) | ... | (EDMA\_TPCC\_IPRH[31] E63 & EDMA\_TPCC\_IERH[31] E63 & EDMA\_TPCC\_DRAEHM\_k.DRAEHM\_0[31] E63)
- EDMA\_TPCC\_INT1: (EDMA\_TPCC\_IPR[0] E0 & EDMA\_TPCC\_IER[0] E0 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_1[0] E0) | (EDMA\_TPCC\_IPR[1] E1 & EDMA\_TPCC\_IER[1] E1 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_1[1] E1) | ... | (EDMA\_TPCC\_IPRH[31] E63 & EDMA\_TPCC\_IERH[31] E63 & EDMA\_TPCC\_DRAEHM\_k.DRAEHM\_1[31] E63)
- EDMA\_TPCC\_INT2: (EDMA\_TPCC\_IPR[0] E0 & EDMA\_TPCC\_IER[0] E0 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_2[0] E0) | (EDMA\_TPCC\_IPR[1] E1 & EDMA\_TPCC\_IER[1] E1 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_2[1] E1) | ... | (EDMA\_TPCC\_IPRH[31] E63 & EDMA\_TPCC\_IERH[31] E63 & EDMA\_TPCC\_DRAEHM\_k.DRAEHM\_2[31] E63)...
- Up to EDMA\_TPCC\_INT7: (EDMA\_TPCC\_IPR[0] E0 & EDMA\_TPCC\_IER[0] E0 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_7[0] E0) | (EDMA\_TPCC\_IPR[1] E1 & EDMA\_TPCC\_IER[1] E1 & EDMA\_TPCC\_DRAEM\_k.DRAEM\_7[1] E1) | ... | (EDMA\_TPCC\_IPRH[31] E63 & EDMA\_TPCC\_IERH[31] E63 & EDMA\_TPCC\_DRAEHM\_k.DRAEHM\_7[31] E63)

### Note

The EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k for all regions are expected to be set up at system initialization and to remain static for an extended period of time. The interrupt enable registers are used for dynamic enable/disable of individual interrupts.

Because there is no relation between the EDMA\_TPCC\_OPT\_n[17:12] TCC value and the DMA/QDMA channel, it is possible, the DMA channel 0 to have the EDMA\_TPCC\_OPT\_n[17:12] TCC = 63 in its associated PaRAM set. This mean that if a transfer completion interrupt is enabled (EDMA\_TPCC\_OPT\_n[20] TCINTEN or EDMA\_TPCC\_OPT\_n[21] ITCINTEN is set), then based on the TCC value, EDMA\_TPCC\_IPRH[31] E63 is set up on completion. For proper channel operations and interrupt generation using the shadow region map - program the EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k that is associated with the shadow region to have read/write access to both bit 0 (corresponding to channel 0) and bit 63 (corresponding to EDMA\_TPCC\_IPRH bit that is set upon completion).

#### 11.3.9.1.2 Clearing Transfer Completion Interrupts

Transfer completion interrupts that are latched to the interrupt pending registers ( EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH ) are cleared by writing a 1 to the corresponding bit in the interrupt pending clear register ( EDMA\_TPCC\_ICR / EDMA\_TPCC\_ICRH ). For example, a write of 1 to EDMA\_TPCC\_ICR[0] E0 clears a pending interrupt in EDMA\_TPCC\_IPR[0] E0.

If an incoming transfer completion code TCC (EDMA\_TPCC\_OPT\_n[17:12] TCC) gets latched to a bit in EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH, then additional bits that get set due to a subsequent transfer completion does not result in asserting the EDMA\_TPCC completion interrupt. In order for the completion interrupt to be pulsed, the required transition is from a state where no enabled interrupts are set to a state where at least one enabled interrupt is set.

#### 11.3.9.2 EDMA Interrupt Servicing

Upon completion of a transfer (early or normal completion), the EDMA channel controller sets the appropriate bit in the interrupt pending registers ( EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH ), as the transfer completion codes specify. If the completion interrupts are appropriately enabled, then the CPU enters the interrupt service routine (ISR) when the completion interrupt is asserted.

After servicing the interrupt, the ISR should clear the corresponding bit in EDMA\_TPCC\_IPR/ EDMA\_TPCC\_IPRH, thereby enabling recognition of future interrupts. The EDMA\_TPCC only asserts additional completion interrupts when all EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH bits clear.

When one interrupt is serviced many other transfer completions may result in additional bits being set in EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH, thereby resulting in additional interrupts. Each of the bits in EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH may need different types of service therefore, the ISR must check all pending interrupts and continue until all of the posted interrupts are serviced appropriately.

Examples of pseudo code for a CPU interrupt service routine for an EDMA\_TPCC completion interrupt are shown in [Example 11-2](#) and [Example 11-3](#).

The ISR routine in [Example 11-2](#) is more exhaustive and incurs a higher latency.

#### Example 11-2. Interrupt Servicing

The pseudo code:

1. Reads the interrupt pending register EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH.
2. Performs the operations needed.
3. Writes to the interrupt pending clear register EDMA\_TPCC\_ICR / EDMA\_TPCC\_ICRH to clear the corresponding EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH bit(s).
4. Reads EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH again:

- a. If `EDMA_TPCC_IPR / EDMA_TPCC_IPRH` is not equal to 0, repeat from step 2 (implies occurrence of new event between step 2 to step 4).
- b. If `EDMA_TPCC_IPR / EDMA_TPCC_IPRH` is equal to 0, assure that all of the enabled interrupts are inactive.

#### Note

An event may occur during step 4 while the `EDMA_TPCC_IPR / EDMA_TPCC_IPRH` bits are read as 0 and the application is still in the interrupt service routine. If this happens, a new interrupt is recorded in the device interrupt controller and a new interrupt generates as soon as the application exits in the interrupt service routine.

#### 11.3.9.3

[Example 11-3](#) is less rigorous, with less burden on the software in polling for set interrupt bits, but can occasionally cause a race condition as mentioned above.

#### Example 11-3. Interrupt Servicing

If any enabled and pending (possibly lower priority) interrupts are left, force the interrupt logic to reassert the interrupt pulse by setting the `EDMA_TPCC_IEVAL[0]` EVAL bit in the interrupt evaluation register.

The pseudo code is as follows:

1. Enters ISR.
2. Reads `EDMA_TPCC_IPR / EDMA_TPCC_IPRH`.
3. For the condition that is set in `EDMA_TPCC_IPR / EDMA_TPCC_IPRH`:
  - a. Service interrupt as the application requires.
  - b. Clear the bit for serviced conditions (others may still be set, and other transfers may have resulted in returning the TCC to `EDMA_TPCC` after step 2).
4. Reads `EDMA_TPCC_IPR / EDMA_TPCC_IPRH` prior to exiting the ISR:
  - a. If `EDMA_TPCC_IPR / EDMA_TPCC_IPRH` is equal to 0, then exit the ISR.
  - b. If `EDMA_TPCC_IPR / EDMA_TPCC_IPRH` is not equal to 0, then set `EDMA_TPCC_IEVAL` so that upon exit of ISR, a new interrupt triggers if any enabled interrupts are still pending.

#### 11.3.9.4 Interrupt Evaluation Operations

The `EDMA_TPCC` has interrupt evaluate registers `EDMA_TPCC_IEVAL` that exist in the global region and in each shadow region. The registers in the shadow region are the only registers in the DMA channel shadow region memory map that are not affected by the settings for the DMA region access enable registers `EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k`. Writing a 1 to the `EDMA_TPCC_IEVAL[0]` EVAL bit in the registers that are associated with a particular shadow region results in pulsing the associated region interrupt (global or shadow), if any enabled interrupt (via `EDMA_TPCC_IER / EDMA_TPCC_IERH`) is still pending `EDMA_TPCC_IPR / EDMA_TPCC_IPRH`. This register assures that the CPU does not miss the interrupts (or the EDMA controller associated with the shadow region) if the software architecture chooses not to use all interrupts. Refer to [Example 11-3](#) about the use of `EDMA_TPCC_IEVAL` in the EDMA interrupt service routine (ISR).

Similarly an error evaluation register `EDMA_TPCC_EEVAL` exists in the global region. Writing a 1 to the `EDMA_TPCC_EEVAL[0]` EVAL bit causes the pulsing of the error interrupt if any pending errors are in `EDMA_TPCC_EMR / EDMA_TPCC_EMRH`, `EDMA_TPCC_QEMR`, or `EDMA_TPCC_CCERR`. See [Section 11.3.9.5 Error Interrupts](#) for additional information regarding error interrupts.

---

**Note**

While using EDMA\_TPCC\_IEVAL for shadow region completion interrupts, check that the EDMA\_TPCC\_IEVAL operated upon is from that particular shadow region memory map.

---

### 11.3.9.5 Error Interrupts

The EDMA\_TPCC error registers provide the capability to differentiate error conditions (event missed, threshold exceed, etc.). Additionally, setting the error bits in these registers results in asserting the EDMA\_TPCC error interrupt. If the EDMA\_TPCC error interrupt is enabled in the device interrupt controller(s), then it allows the CPU(s) to handle the error conditions.

The EDMA\_TPCC has a single error interrupt (EDMA\_TPCC\_ERRINT) that is asserted for all EDMA\_TPCC error conditions. There are four conditions that cause the error interrupt:

- DMA missed events: for all 64 DMA channels. DMA missed events are latched in the event missed registers EDMA\_TPCC\_EMR / EDMA\_TPCC\_EMRH.
- QDMA missed events: for all 8 QDMA channels. QDMA missed events are latched in the QDMA event missed register EDMA\_TPCC\_QEMR.
- Threshold exceed: for all event queues. These are latched in EDMA\_TPCC error register EDMA\_TPCC\_CCERR.
- TCC error: for outstanding transfer requests that are expected to return completion code EDMA\_TPCC\_OPT\_n[22] TCCHEN or EDMA\_TPCC\_OPT\_n[23] TCINTEN bit is set to 1, exceeding the maximum limit of 63. This is also latched in the EDMA\_TPCC error register EDMA\_TPCC\_CCERR.

Figure 11-17 illustrates the EDMA\_TPCC error interrupt generation operation.

If any of the bits are set in the error registers due to any error condition, the EDMA\_TPCC\_ERRINT is always asserted, as there are no enables for masking these error events. Similar to transfer completion interrupts (EDMA\_TPCC\_INT), the error interrupt also only pulses when the error interrupt condition transitions from no errors being set to at least one error being set. If additional error events are latched prior to the original error bits clearing, the EDMA\_TPCC does not generate additional interrupt.

To reduce the burden on the software, there is an error evaluate register EDMA\_TPCC\_EEVAL that allows re-evaluation of pending set error events/bits, similar to the interrupt evaluate register EDMA\_TPCC\_IEVAL. Unlike the EDMA\_TPCC\_IEVAL functionality, the EDMA\_TPCC\_EEVAL register must be written with '1' after any error interrupts are serviced (even when all pending errors are cleared) in order for subsequent errors to trigger a new interrupt.

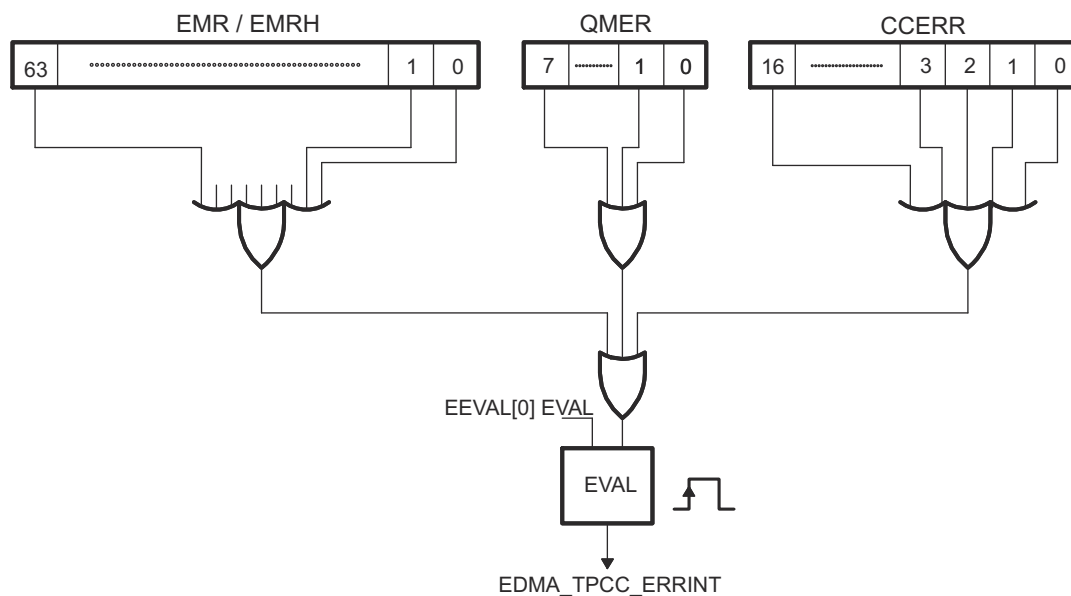
---

#### Note

It is good practice to enable the error interrupt in the device interrupt controller and to associate an interrupt service routine with it to address the various error conditions appropriately. Doing so puts less burden on the software (polling for error status), it provides a good debug mechanism for unexpected error conditions.

---





edma-017

Figure 11-17. Error Interrupt Operation

### 11.3.10 Memory Protection

The EDMA channel controller supports two kinds of memory protection: active and proxy.

#### 11.3.10.1 Active Memory Protection

Active memory protection is a feature that allows or prevents read and write accesses to the EDMA\_TPCC registers. Active memory protection is achieved by a set of memory protection permissions attribute EDMA\_TPCC\_MPPAN\_k registers.

The EDMA\_TPCC register map is divided into three categories:

- a global region.
- a global channel region.
- eight shadow regions.

Each shadow region consists of the respective shadow region registers and the associated PaRAM. For more detailed information regarding the contents of a shadow region, refer to the associated Register Addendum.

Each of the eight shadow regions has an associated EDMA\_TPCC\_MPPAN\_k registers that defines the specific requestor(s) and types of requests that are allowed to the regions resources.

The global channel region is also protected with a memory-mapped register EDMA\_TPCC\_MPPAG. The EDMA\_TPCC\_MPPAG applies to the global region and to the global channel region, except the other EDMA\_TPCC\_MPPAN\_k registers themselves.

Table 11-22 shows the accesses that are allowed or not allowed to the EDMA\_TPCC\_MPPAG and EDMA\_TPCC\_MPPAN\_k. The active memory protection uses the EDMA\_TPCC\_OPT\_n[31] PRIV and EDMA\_TPCC\_OPT\_n[27:24] PRIVID attributes of the EDMA peripheral modules. The EDMA\_TPCC\_OPT\_n[31] PRIV is the privilege level (i.e., user vs. supervisor).

The EDMA\_TPCC\_OPT\_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an EDMA peripheral modules.

**Table 11-22. Allowed Accesses**

Access	Supervisor	User
Read	Yes	Yes
Write	Yes	No

Table 11-23 describes the EDMA\_TPCC\_MPPAN\_k register mapping for the shadow regions (which includes shadow region registers and PaRAM addresses).

The region-based EDMA\_TPCC\_MPPAN\_k registers are used to protect accesses to the DMA shadow regions and the associated region PaRAM. Because there are eight regions, there are eight EDMA\_TPCC\_MPPAN\_k region registers (MPPAN[0-7]).

**Table 11-23. MPPA Registers to Region Assignment**

Register	Registers Protect	Address Range	PaRAM Protect <sup>(1)</sup>	Address Range
EDMA_TPCC_MPPAG	Global Range	0000h-1FFCh	N/A	N/A
EDMA_TPCC_MPPAN_k. MPPAN_0	DMA Shadow 0	2000h-21FCh	1st octant	4000h-47FCh
MPPAN_1	DMA Shadow 1	2200h-23FCh	2nd octant	4800h-4FFCh
MPPAN_2	DMA Shadow 2	2400h-25FCh	3rd octant	5000h-57FCh
MPPAN_3	DMA Shadow 3	2600h-27FCh	4th octant	5800h-5FFCh
MPPAN_4	DMA Shadow 4	2800h-29FCh	5th octant	6000h-67FCh
MPPAN_5	DMA Shadow 5	2A00h-2BFCh	6th octant	6800h-6FFCh
MPPAN_6	DMA Shadow 6	2C00h-2DFCh	7th octant	7000h-77FCh

**Table 11-23. MPPA Registers to Region Assignment (continued)**

Register	Registers Protect	Address Range	PaRAM Protect <sup>(1)</sup>	Address Range
MPPAN_7	DMA Shadow 7	2E00h-2FFCh	8th octant	7800h-7FFCh

(1) The PARAM region is divided into 8 regions referred to as an octant.

#### Example Access denied.

Write access to shadow region 7's event enable set register EDMA\_TPCC\_EESR:

1. The original value of the event enable register EDMA\_TPCC\_EER at address offset 0x1020 is 0x0.
2. The EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[7] NS is set to prevent user level accesses (EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[1] UW = 0, EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[2] UR = 0), but it allows supervisor level accesses (EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[4] SW = 1, EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[5] SR = 1) with a privilege ID of 0. (EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0 attempts to perform a user-level write of a value of 0xFF00FF00 to shadow region 7's event enable set register EDMA\_TPCC\_EESR at address offset 0x2E30.

#### Note

The EDMA\_TPCC\_EER is a read-only register and the only way that write to it is by writing to the EDMA\_TPCC\_EESR. There is only one physical register for EDMA\_TPCC\_EER, EDMA\_TPCC\_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[1] UW = 0, though the privilege ID of the write access is set to 0, the access is not allowed and the EDMA\_TPCC\_EER is not written too.

#### Example Access Allowed

Write access to shadow region 7's event enable set register EDMA\_TPCC\_EESR:

1. The original value of the event enable register EDMA\_TPCC\_EER at address offset 0x1020 is 0x0.
2. The EDMA\_TPCC\_MPPAN\_k.EDMA\_TPCC\_MPPAN\_7 is set to allow user-level accesses (EDMA\_TPCC\_MPPAN\_k.EDMA\_TPCC\_MPPAN\_7[1] UW = 1, EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[2] UR = 1) and supervisor-level accesses (EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[4] SW = 1, EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[5] SR = 1) with a privilege ID of 0. (EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0, attempts to perform a user-level write of a value of 0xABCD0123 to shadow region 7's event enable set register EDMA\_TPCC\_EESR at address offset 0x2E30.

#### Note

The EDMA\_TPCC\_EER is a read-only register and the only way that write to it is by writing to the EDMA\_TPCC\_EESR. There is only one physical register for EDMA\_TPCC\_EER, EDMA\_TPCC\_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA\_TPCC\_MPPAN\_k. EDMA\_TPCC\_MPPAN\_7[1] UW = 1 and EDMA\_TPCC\_MPPAN\_k. MPPAN\_7[10] AID0 = 1, the user-level write access is allowed.
5. The accesses to shadow region registers are masked by their respective EDMA\_TPCC\_DRAEM\_k register. In this example, the EDMA\_TPCC\_DRAEM\_k. EDMA\_TPCC\_DRAEM\_7 is set of 0x9FF00FC2.
6. The value finally written to EDMA\_TPCC\_EER is 0x8BC00102.

#### 11.3.10.2 Proxy Memory Protection

Proxy memory protection allows an EDMA transfer programmed by a given peripheral module connected to EDMA, to have its permissions travel with the transfer through the EDMA\_TPTC. The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints. The

EDMA\_TPCC\_OPT\_n[31] PRIV bit and EDMA\_TPCC\_OPT\_n[27:24] PRIVID bit is set with the peripheral module's PRIV value and PRIVID values, respectively, when any part of the PaRAM set is written.

The EDMA\_TPCC\_OPT\_n[31] PRIV is the privilege level (i.e., user vs. supervisor). The EDMA\_TPCC\_OPT\_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an peripheral module connected to EDMA.

These options are part of the TR that are submitted to the transfer controller. The transfer controller uses the above values on their respective read and write command bus so that the target endpoints can perform memory protection checks based on these values.

Consider a parameter set that is programmed by a CPU in user privilege level for a simple transfer with the source buffer on an L2 page and the destination buffer on an L1D page. The EDMA\_TPCC\_OPT\_n[31] PRIV is 0 for user-level and the CPU has a EDMA\_TPCC\_OPT\_n[27:24] PRIVID to 0.

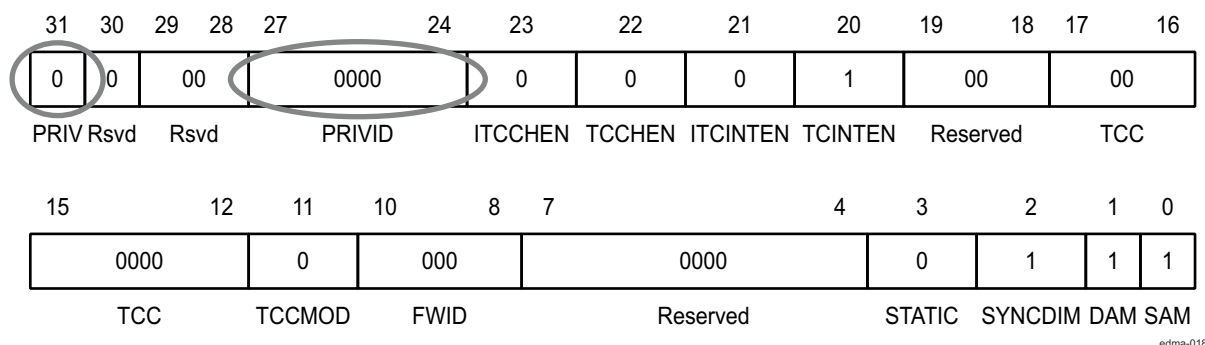
The PaRAM set is shown in Figure 11-18.

**Figure 11-18. PaRAM Set Content for Proxy Memory Protection Example**

(a) EDMA Parameters

Parameter Contents				Parameter	
	0010 0007h			Channel Options Parameter (OPT)	
	009F 0000h			Channel Source Address (SRC)	
	0001h	0004h		Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
	00F0 7800h			Channel Destination Address (DST)	
	0001h	0001h		Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
	0000h	FFFFh		BCNT Reload (BCNTRLD)	Link Address (LINK)
	0001h	1000h		Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
	0000h	0001h		Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT\_n) Content



edma-018

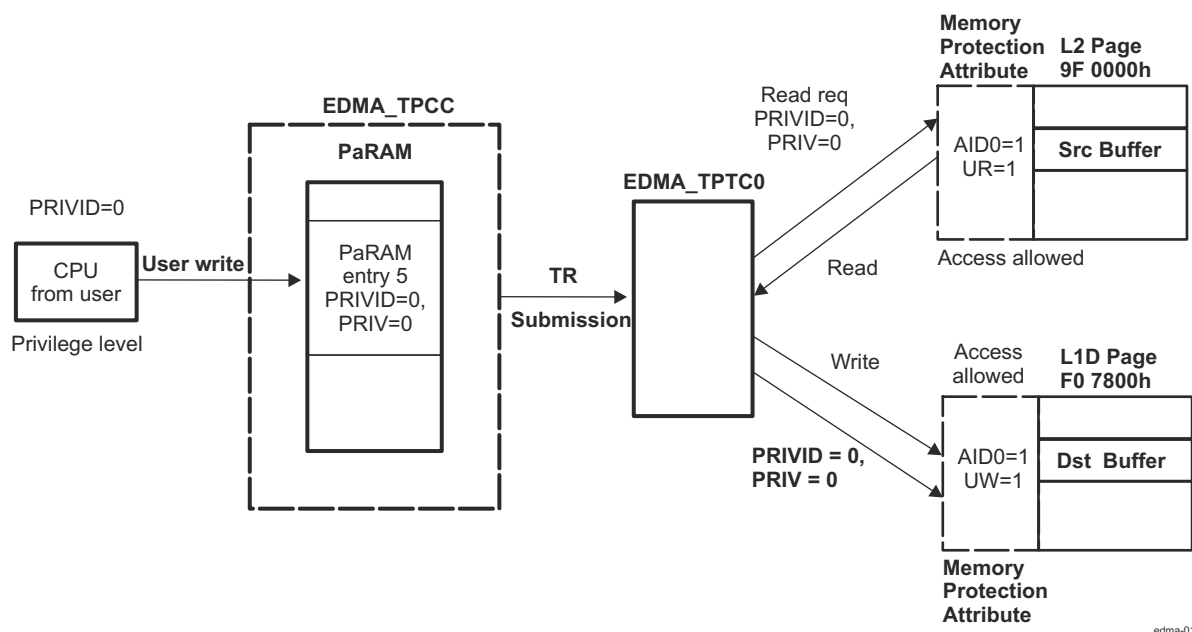
**Figure 11-19. Channel Options Parameter (OPT) Example**

The EDMA\_TPCC\_OPT\_n[31] PRIV and EDMA\_TPCC\_OPT\_n[27:24] PRIVID information travels along with the read and write requests that are issued to the source and destination memories.

For example, if the access attributes that are associated with the L2 page with the source buffer only allow supervisor read, write accesses (EDMA\_TPCC\_MPPAN\_k[4] SW and EDMA\_TPCC\_MPPAN\_k[5] SR), the user-level read request above is refused. Similarly, if the access attributes that are associated with the L1D page with the destination buffer only allow supervisor read and write accesses (EDMA\_TPCC\_MPPAN\_k[4] SW, EDMA\_TPCC\_MPPAN\_k[5] SR), the user-level write request above is refused. For the transfer to succeed, the source and destination pages must have user-read and user-write permissions, respectively, along with allowing accesses from a PRIVID = 0.

Because the privilege level and privilege identification travel with the read and write requests, EDMA acts as a proxy.

Figure 11-20 illustrates the propagation of EDMA\_TPCC\_OPT\_n[31] PRIV and EDMA\_TPCC\_OPT\_n[27:24] PRIVID at the boundaries of all the interacting entities (CPU, EDMA\_TPCC, EDMA\_TPTCs, and target memories).



**Figure 11-20. Proxy Memory Protection Example**

### 11.3.11 Event Queue(s)

Event queues are a part of the EDMA channel controller. Event queues form the interface between the event detection logic in the EDMA\_TPCC and the transfer request (TR) submission logic of the EDMA\_TPCC. Each queue is 16 entries deep. Each event queue can queue a maximum of 16 events. If there are more than 16 events, then the events that cannot find a place in the event queue remain set in the associated event register and the CPU does not stall.

There are two event queues for the device: Queue0, Queue1. Events in Queue0 result in submission of its associated transfer requests (TRs) to TC0. The transfer requests that are associated with events in Queue1 are submitted to TC1.

An event that wins prioritization against other DMA and/or QDMA pending events is placed at the tail of the appropriate event queue. Each event queue is serviced in FIFO order. Once the event reaches the head of its queue and the corresponding transfer controller is ready to receive another TR, the event is de-queued and the PaRAM set corresponding to the de-queued event is processed and submitted as a transfer request packet (TRP) to the associated EDMA transfer controller.

Queue0 has highest priority and Queue1 has the lowest priority, if Queue0 and Queue1 both have at least one event entry and if both TC0 and TC1 can accept transfer requests, then the event in Queue0 is de-queued first and its associated PaRAM set is processed and submitted as a transfer request (TR) to TC0.

Refer to *Performance Considerations* for system-level performance considerations. All of the event entries in all of the event queues are software readable (not writeable) by accessing the event entry registers EDMA\_TPCC\_Q0E\_p and EDMA\_TPCC\_Q1E\_p. Each event entry register characterizes the queued event in terms of the type of event (manual, event, chained or auto-triggered) and the event number. Refer to the associated Register Addendum for EDMA\_TPCC\_Q0E\_p / EDMA\_TPCC\_Q1E\_p descriptions of the bit fields.

### 11.3.11.1 DMA/QDMA Channel to Event Queue Mapping

Each of the 64 DMA channels and eight QDMA channels are programmed independently to map to a specific queue, using the DMA queue number register EDMA\_TPCC\_DMAQNUMN\_k and the QDMA queue number register EDMA\_TPCC\_QDMAQNUM. The mapping of DMA/QDMA channels is critical to achieving the desired performance level for the EDMA and most importantly, in meeting real-time deadlines. Refer to *System-level Performance Considerations*.

#### Note

If an event is ready to be queued and both the event queue and the EDMA transfer controller that is associated to the event queue are empty, then the event bypasses the event queue, and moves the PaRAM processing logic, and eventually to the transfer request submission logic for submission to the EDMA\_TPTC. In this case, the event is not logged in the event queue status registers.

### 11.3.11.2 Queue RAM Debug Visibility

There are two event queues and each queue has 16 entries. These 16 entries are managed in a circular FIFO. There is a queue status register EDMA\_TPCC\_QSTATN\_i associated with each queue. These along with all of the 16 entries per queue can be read via registers EDMA\_TPCC\_QSTATN\_i and Q0E\_p / Q1E\_p, respectively.

These registers provide user visibility.

The event queue entry register (QxEy Q0E\_p / Q1E\_p) uniquely identifies the specific event type (event-triggered, manually-triggered, chain-triggered, and QDMA events) along with the event number (for all DMA/QDMA event channels) that are in the queue or have been de-queued (passed through the queue).

Each of the 16 entries in the event queue are read using the EDMA\_TPCC memory-mapped register. To see the history of the last 16 TRs that have been processed by the EDMA on a given queue, read the event queue registers. This provides user/software visibility and is helpful for debugging real-time issues (typically post-mortem), involving multiple events and event sources.

The queue status register (QSTATn EDMA\_TPCC\_QSTATN\_i) includes fields for the start pointer EDMA\_TPCC\_QSTATN\_i[3:0] STRTPTR which provides the offset to the head entry of an event. It also includes a field called EDMA\_TPCC\_QSTATN\_i[12:8] NUMVAL that provides the total number of valid entries residing in the event queue at a given instance of time. The EDMA\_TPCC\_QSTATN\_i[3:0] STRTPTR is used to index appropriately into the 16 event entries. EDMA\_TPCC\_QSTATN\_i[12:8] NUMVAL number of entries starting from STRTPTR are indicative of events still queued in the respective queue. The remaining entry must be read to determine what's already de-queued and submitted to the associated transfer controller.

### 11.3.11.3 Queue Resource Tracking

The EDMA\_TPCC event queue includes watermarking/threshold logic that allows to keep track of maximum usage of all event queues. This is useful for debugging real-time deadline violations that may result from head-of-line blocking on a given EDMA event queue.

The maximum number of events are programmed that the queue up in an event queue by programming the threshold value (between 0 to 15) in the queue watermark threshold A register EDMA\_TPCC\_QWMTHRA. The maximum queue usage is recorded actively in the watermark EDMA\_TPCC\_QSTATN\_i[20:16] WM field of the queue status register, that keeps getting updated based on a comparison of number of valid entries, which is also visible in the EDMA\_TPCC\_QSTATN\_i[12:8] NUMVAL bit and the maximum number of entries.

If the queue usage is exceeded, this status is visible in the EDMA\_TPCC registers: the QTHRXCDn bits in the channel controller error register EDMA\_TPCC\_CCERR[7:0] and the EDMA\_TPCC\_QSTATN\_i[24] THRXCD bit, where n stands for the event queue number. Any bits that are set in EDMA\_TPCC\_CCERR also generate an EDMA\_TPCC error interrupt.

### 11.3.11.4 Performance Considerations

The device system bus infrastructure arbitrates bus requests from all of the controllers (TCs, CPU(S), and other bus controllers) to the shared target resources (peripherals and memories).

Therefore, the priority of unloading queues has a secondary affect compared to the priority of the transfers as they are executed by the EDMA\_TPTC.

### 11.3.12 EDMA Transfer Controller (EDMA\_TPTC)

The EDMA channel controller is the user-interface of the EDMA and the EDMA transfer controller (EDMA\_TPTC) is the data movement engine of the EDMA controller. The EDMA\_TPCC submits transfer requests (TR) to the EDMA\_TPTC and the EDMA\_TPTC performs the data transfers dictated by the TR, so the EDMA\_TPTC is a target to the EDMA\_TPCC.

#### 11.3.12.1 Architecture Details

##### 11.3.12.1.1 Command Fragmentation

The TC read and write controllers in conjunction with the source and destination register sets are responsible for issuing optimally-sized reads and writes to the target endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS), which is defined in the *TPTC DBS Configuration registers*.

The EDMA\_TPTC attempts to issue the largest possible command size as limited by the DBS value or the EDMA\_TPCC\_ABCNT\_n[15:0] ACNT and EDMA\_TPCC\_ABCNT\_n[31:16] BCNT value of the TR. EDMA\_TPTC obeys the following rules:

- The read/write controllers always issue commands less than or equal to the DBS value.
- The first command of a 1D transfer command always aligns the address of subsequent commands to the DBS value.

Table 11-24 lists the TR segmentation rules that are followed by the EDMA\_TPTC. In summary, if the EDMA\_TPCC\_ABCNT\_n[15:0] ACNT value is larger than the DBS value, then the EDMA\_TPTC breaks the EDMA\_TPCC\_ABCNT\_n[15:0] ACNT array into DBS-sized commands to the source/destination addresses. Each EDMA\_TPCC\_ABCNT\_n[31:16] BCNT number of arrays are then serviced in succession.

For BCNT arrays of ACNT bytes (that is, a 2D transfer), if the EDMA\_TPCC\_ABCNT\_n[15:0] ACNT value is less than or equal to the DBS value, then the TR may be optimized into a 1D-transfer in order to maximize efficiency. The optimization takes place if the EDMA\_TPTC recognizes that the 2D-transfer is organized as a single dimension (EDMA\_TPCC\_ABCNT\_n[15:0] ACNT == EDMA\_TPCC\_BIDX\_n) and the ACNT value is a power of 2.

Table 11-24 lists conditions in which the optimizations are performed.

**Table 11-24. Read/Write Command Optimization Rules**

ACNT ≤ DBS	ACNT is power of 2	BIDX = ACNT	BCNT ≤ 1023	SAM/DAM = Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	x	x	x	x	Not Optimized
x	No	x	x	x	Not Optimized
x	x	No	x	x	Not Optimized
x	x	x	No	x	Not Optimized
x	x	x	x	No	Not Optimized

##### 11.3.12.1.2 TR Pipelining

TR pipelining refers to the ability of the source active set to proceed ahead of the destination active set. Essentially, the reads for a given TR may already be in progress while the writes of a previous TR may not have completed.

The number of outstanding TRs is limited by the number of destination FIFO register entries.

TR pipelining is useful for maintaining throughput on back-to-back small TRs. It minimizes the startup overhead because reads start in the background of a previous TR writes.



### Example 11-4. Command Fragmentation (DBS = 64)

The pseudo code:

1. EDMA\_TPTCn\_PCNT[15:0] ACNT = 8, EDMA\_TPTCn\_PCNT[31:16] BCNT = 8,  
EDMA\_TPTCn\_PBDX[15:0] SBIDX = 8, EDMA\_TPTCn\_PBDX[31:16] DBIDX = 10,  
EDMA\_TPTCn\_PSRC[31:0] SADDR = 64, EDMA\_TPTCn\_SADST[31:0] DADDR = 191

Read Controller: This is optimized from a 2D-transfer to a 1D-transfer such that the read side is equivalent to EDMA\_TPTCn\_PCNT[15:0] ACNT = 64, EDMA\_TPTCn\_PCNT[31:16] BCNT = 1.

Cmd0 = 64 byte

Write Controller: Because DBIDX != ACNT, it is not optimized.

Cmd0 = 8 byte, Cmd1 = 8 byte, Cmd2 = 8 byte, Cmd3 = 8 byte, Cmd4 = 8 byte, Cmd5 = 8 byte, Cmd6 = 8 byte, Cmd7 = 8 byte.

2. EDMA\_TPTCn\_PCNT[15:0] ACNT=128, EDMA\_TPTCn\_PCNT[31:16] BCNT = 1,  
EDMA\_TPTCn\_PSRC[31:0] SADDR = 63, EDMA\_TPTCn\_SADST[31:0] DADDR = 513

Read Controller: Read address is not aligned.

Cmd0 = 1 byte, (now the SADDR is aligned to 64 for the next command)

Cmd1 = 64 bytes

Cmd2 = 63 bytes

Write Controller: The write address is also not aligned.

Cmd0 = 63 bytes, (now the DADDR is aligned to 64 for the next command)

Cmd1 = 64 bytes

Cmd2 = 1 byte

#### 11.3.12.1.3 Performance Tuning

By default, reads are as issued as fast as possible. In some cases, the reads issued by the EDMA\_TPTC could fill the available command buffering for a target, delaying other (potentially higher priority) controllers from successfully submitting commands to that target. The rate at which read commands are issued by the EDMA\_TPTC is controlled by the EDMA\_TPTCn\_RDRATE register. The EDMA\_TPTCn\_RDRATE register defines the number of cycles that the EDMA\_TPTC read controller waits before issuing subsequent commands for a given TR, thus minimizing the chance of the EDMA\_TPTC consuming all available target resources. The EDMA\_TPTCn\_RDRATE[2:0] RDRATE value must be set to a relatively small value if the transfer controller is targeted for high priority transfers and to a higher value if the transfer controller is targeted for low priority transfers.

In contrast, the Write Interface does not have any performance turning knobs because writes always have an interval between commands as write commands are submitted along with the associated write data.

#### 11.3.12.2 Memory Protection

The transfer controller plays an important role in handling proxy memory protection. There are two access properties associated with a transfer: for instance, the privilege id (system-wide identification assigned to a controller) of the controller initiating the transfer, and the privilege level (user versus supervisor) used to program the transfer. This information is maintained in the PaRAM set when it is programmed in the channel controller. When a TR is submitted to the transfer controller, this information is made available to the EDMA\_TPTC and used by the EDMA\_TPTC while issuing read and write commands. The read or write commands have the same privilege identification, and privilege level as that programmed in the EDMA transfer in the channel controller.



### 11.3.12.3 Error Generation

Errors are generated if enabled under three conditions:

- EDMA\_TPTC detection of an error signaled by the source or destination address.
- Attempt to read or write to an invalid address in the configuration memory map.
- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source/destination addresses and source/destination indexes must be aligned to 32 bytes).

Either or all error types may be disabled. If an error bit is set and enabled, the error interrupt for the concerned transfer controller is generated.

### 11.3.12.4 Debug Features

The DMA program register set, DMA source active register set, and the destination FIFO register set are used to derive a brief history of TRs serviced through the transfer controller.

Additionally, the EDMA\_TPTC status register EDMA\_TPTCn\_TCSTAT has dedicated bit fields to indicate the ongoing activity within different parts of the transfer controller:

- The EDMA\_TPTCn\_TCSTAT[1] SRCACTV bit indicates whether the source active set is active.
- The EDMA\_TPTCn\_TCSTAT[6:4] DSTACTV bit indicates the number of TRs resident in the destination register active set at a given instance.
- The EDMA\_TPTCn\_TCSTAT[0] PROGBUSY bit indicates whether a valid TR is present in the DMA program set.

---

#### Note

If the TRs are in progression, it must realize that there is a chance that the values read from the EDMA\_TPTC status registers will be inconsistent since the EDMA\_TPTC changes the values of these registers due to ongoing activities.

It is recommended that to ensure no additional submission of TRs to the EDMA\_TPTC in order to facilitate ease of debug.

---

#### 11.3.12.4.1 Destination FIFO Register Pointer

The destination FIFO register pointer is implemented as a circular buffer with the start pointer being EDMA\_TPTCn\_TCSTAT[12:11] DFSTRTPTR and a buffer depth of usually 2 or 4. The EDMA\_TPTC maintains two important status details in EDMA\_TPTCn\_TCSTAT that are used during advanced debugging, if necessary. The EDMA\_TPTCn\_TCSTAT[12:11] DFSTRTPTR is a start pointer, the index to the head of the destination FIFO register. The EDMA\_TPTCn\_TCSTAT[6:4] DSTACTV is a counter for the number of valid (occupied) entries. These registers are used to get a brief history of transfers.

Examples of some register field values and their interpretation:

- EDMA\_TPTCn\_TCSTAT[12:11] DFSTRTPTR = 0x0 and EDMA\_TPTCn\_TCSTAT[6:4] DSTACTV = 0x0 implies that no TRs are stored in the destination FIFO register.
- EDMA\_TPTCn\_TCSTAT[12:11] DFSTRTPTR = 0x1 and EDMA\_TPTCn\_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 1 and the second pending TR is read from the destination FIFO register entry 2.
- EDMA\_TPTCn\_TCSTAT[12:11] DFSTRTPTR = 0x3 and EDMA\_TPTCn\_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 3 and the second pending TR is read from the destination FIFO register entry 0.

### 11.3.13 Event Dataflow

This section summarizes the data flow of a single event, from the time the event is latched to the channel controller to the time the transfer completion code is returned. The following steps list the sequence of EDMA\_TPCC activity:

1. Event is asserted from an external source (peripheral or external interrupt). This also is similar for a manually-triggered, chained-triggered, or QDMA-triggered event. The event is latched into the `EDMA_TPCC_ER[31:0]En / EDMA_TPCC_ERH[31:0] En` (or `EDMA_TPCC_CER[31:0] En / EDMA_TPCC_CERH[31:0] En`, `EDMA_TPCC_ESR[31:0] En / EDMA_TPCC_ESRH[31:0] En`, `EDMA_TPCC_QER[7:0] En`) bit.
2. Once an event is prioritized and queued into the appropriate event queue, the `EDMA_TPCC_SER[31:0] En \ EDMA_TPCC_SERH[31:0] En` (or `EDMA_TPCC_QSER[7:0] En`) bit is set to inform the event prioritization / processing logic to disregard this event since it is already in the queue. Alternatively, if the transfer controller and the event queue are empty, then the event bypasses the queue.
3. The `EDMA_TPCC` processing and the submission logic evaluates the appropriate PaRAM set and determines whether it is a non-null and non-dummy transfer request (TR).
4. The `EDMA_TPCC` clears the `EDMA_TPCC_ER[31:0] En/ EDMA_TPCC_ERH[31:0] En` (or `EDMA_TPCC_CER[31:0] En / EDMA_TPCC_CERH[31:0] En`, `EDMA_TPCC_ESR[31:0]En / EDMA_TPCC_ESRH[31:0] En`, `EDMA_TPCC_QER[31:0] En`) bit and the `EDMA_TPCC_SER[31:0] En/ EDMA_TPCC_SERH[31:0] En` bit as soon as it determines the TR is non-null. In the case of a null set, the `EDMA_TPCC_SER[31:0] En/ EDMA_TPCC_SERH[31:0] En` bit remains set. It submits the non-null/non-dummy TR to the associated transfer controller. If the TR was programmed for early completion, the `EDMA_TPCC` immediately sets the interrupt pending register (`EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC] - 32`).
5. If the TR was programmed for normal completion, the `EDMA_TPCC` sets the interrupt pending register (`EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC]`) when the `EDMA_TPTC` informs the `EDMA_TPCC` about completion of the transfer (returns transfer completion codes).
6. The `EDMA_TPCC` programs the associated `EDMA_TPTC`'s Program Register Set with the TR.
7. The TR is then passed to the Source Active set and the DST FIFO Register Set, if both the register sets are available.
8. The Read Controller processes the TR by issuing read commands to the source peripheral endpoint. The Read Data lands in the Data FIFO of the `EDMA_TPTCn`.
9. As soon as sufficient data is available, the Write Controller begins processing the TR by issuing write commands to the destination peripheral endpoint.
10. This continues until the TR completes and the `EDMA_TPTCn` then signals completion status to the `EDMA_TPCC`.

### 11.3.14 EDMA Controller Prioritization

The EDMA controller has many implementation rules to deal with concurrent events/channels, transfers, etc. The following subsections detail various arbitration details whenever there might be occurrence of concurrent activity. [Figure 11-21](#) shows the different places EDMA priorities come into play.

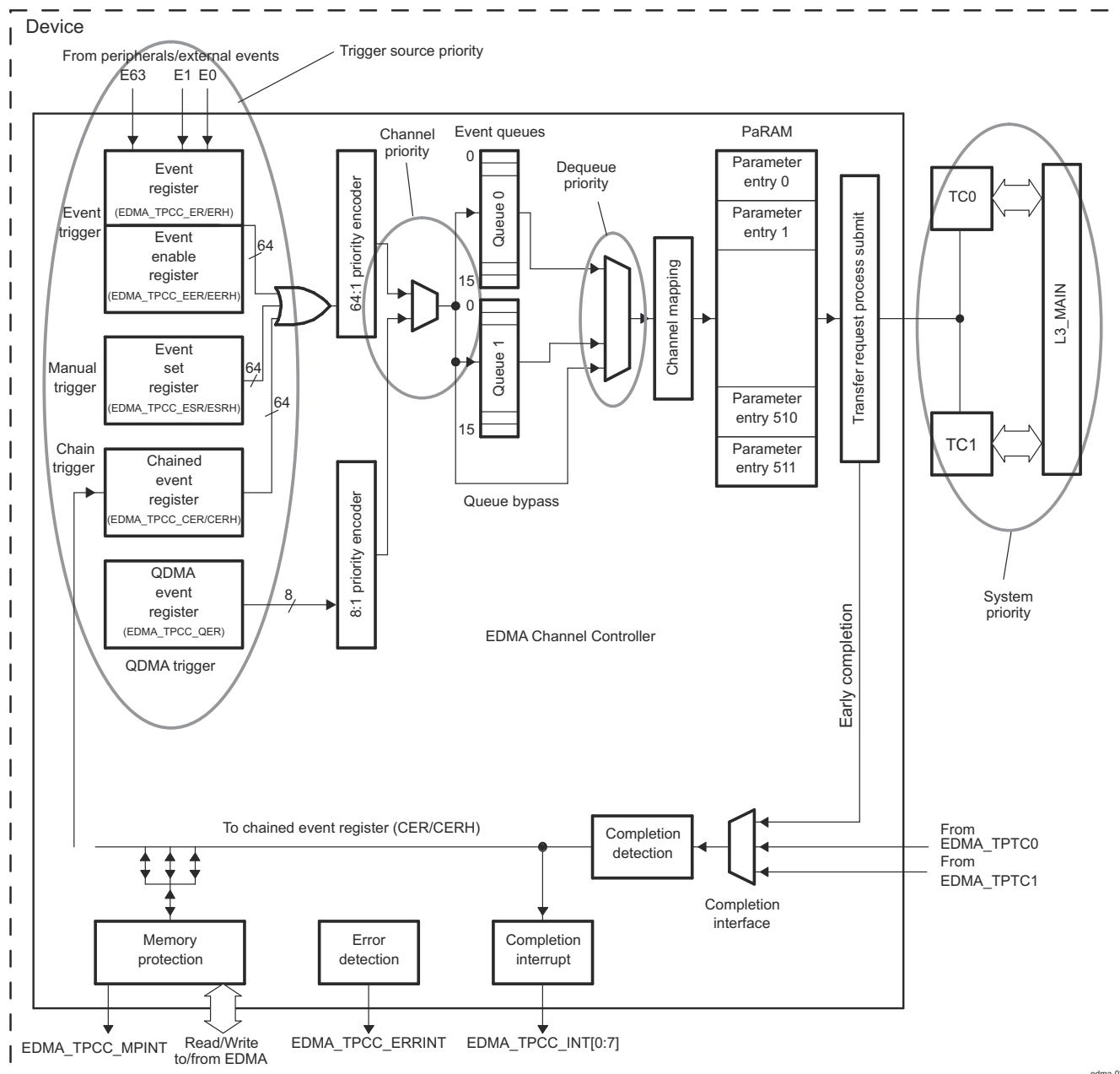


Figure 11-21. EDMA Prioritization

#### 11.3.14.1 Channel Priority

The EDMA event registers EDMA\_TPCC\_ER and EDMA\_TPCC\_ERH capture up to 64 events, the QDMA event register EDMA\_TPCC\_QER captures QDMA events for all QDMA channels therefore, it is possible for events to occur simultaneously on the DMA/QDMA event inputs. For events arriving simultaneously, the event associated with the lowest channel number is prioritized for submission to the event queues (for DMA events, channel 0 has the highest priority and channel 63 has the lowest priority, for QDMA events, channel 0 has the highest priority and channel 7 has the lowest priority). This mechanism only sorts simultaneous events for submission to the event queues.

If a DMA and QDMA event occurs simultaneously, the DMA event always has prioritization against the QDMA event for submission to the event queues.

#### 11.3.14.2 Trigger Source Priority

If a EDMA channel is associated with more than one trigger source (event trigger, manual trigger, and chain trigger), and if multiple events are set simultaneously for the same channel (EDMA\_TPCC\_ER[31:0]  $E_n = 1$ , EDMA\_TPCC\_ESR[31:0]  $E_n = 1$ , EDMA\_TPCC\_CER[31:0]  $E_n = 1$ ), then the EDMA\_TPCC always services these events in the following priority order: event trigger (via EDMA\_TPCC\_ER) is higher priority than chain trigger (via EDMA\_TPCC\_CER) and chain trigger is higher priority than manual trigger (via EDMA\_TPCC\_ESR).

This implies that if for channel 0, both EDMA\_TPCC\_ER[0]  $E_0 = 1$  and EDMA\_TPCC\_CER[0]  $E_0 = 1$  at the same time, then the EDMA\_TPCC\_ER[0]  $E_0$  event is always queued before the EDMA\_TPCC\_CER[0]  $E_0$  event.

#### 11.3.14.3 Dequeue Priority

The priority of the associated transfer request (TR) is further mitigated by which event queue is being used for event submission (dictated by EDMA\_TPCC\_DMAQNUMN\_k and EDMA\_TPCC\_QDMAQNUM). For submission of a TR to the transfer request, events need to be de-queued from the event queues. Queue 0 has the highest dequeue priority and queue 1 the lowest.

#### 11.3.15 Emulation Considerations

During debug when using the emulator, the CPU(s) may be halted on an execute packet boundary for single-stepping, benchmarking, profiling, or other debug purposes. During an emulation halt, the EDMA channel controller and transfer controller operations continue. Events continue to be latched and processed and transfer requests continue to be submitted and serviced.

Since EDMA is involved in servicing multiple controller and target peripherals, it is not feasible to have an independent behavior of the EDMA for emulation halts. EDMA functionality would be coupled with the peripherals it is servicing, which might have different behavior during emulation halts.

## 11.4 EDMA Transfer Examples

The EDMA channel controller performs a variety of transfers depending on the parameter configuration. The following sections provide a description and PaRAM configuration for some typical use case scenarios.

### 11.4.1 Block Move Example

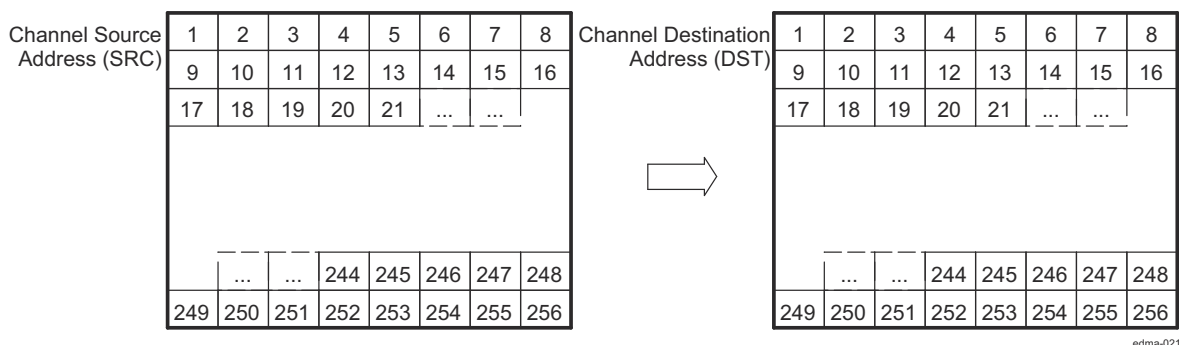
The most basic transfer performed by the EDMA is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM as shown in [Figure 11-22](#).

The source address for the transfer is set to the start of the data block in external memory, and the destination address is set to the start of the data block in L2. If the data block is less than 64K bytes, the PaRAM configuration shown in [Figure 11-23](#) holds true with the synchronization type set to A-synchronized and indexes cleared to 0. If the amount of data is greater than or equal to 64K bytes, EDMA\_TPCC\_ABCNT\_n[31:16] BCNT and the B-indexes need to be set appropriately with the synchronization type set to AB-synchronized. The EDMA\_TPCC\_OPT\_n[3] STATIC bit is set to prevent linking.

This transfer example may also be set up using QDMA. For successive transfer submissions, of a similar nature, the number of cycles used to submit the transfer are fewer depending on the number of changing transfer parameters. The QDMA trigger word must be programmed to be the highest numbered offset in the PaRAM set that undergoes change.

[Figure 11-23](#) shows the parameters Block Move transfer.



**Figure 11-22. Block Move Example**

**Figure 11-23. Block Move Example PaRAM Configuration**

(a) EDMA Parameters

Parameter Contents			Parameter	
0010 0008h			Channel Options Parameter (OPT)	
Channel Source Address (SRC)			Channel Source Address (SRC)	
0001h	FFFFh		Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)			Channel Destination Address (DST)	
0000h	0000h		Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh		BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h		Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h		Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

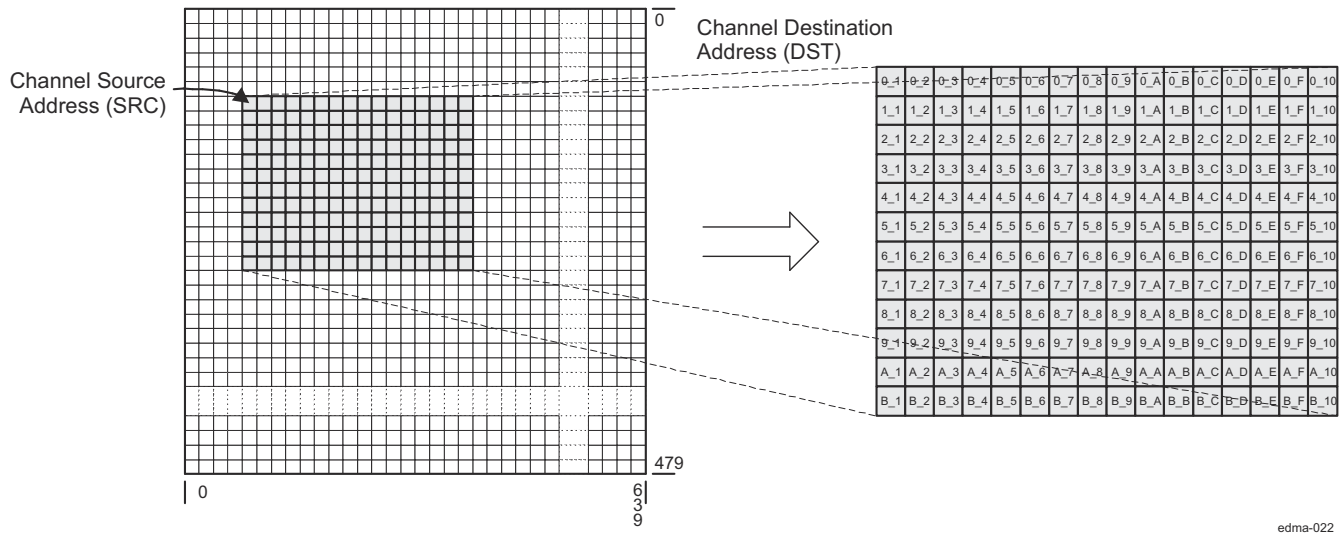
- EDMA\_TPCC\_OPT\_n[3] STATIC = 0x1
- EDMA\_TPCC\_OPT\_n[20] TCINTEN = 0x1

### 11.4.2 Subframe Extraction Example

The EDMA can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory. Each pixel is represented by a 16-bit halfword. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA places the subframe in internal L2 SRAM. Figure 11-24 shows the transfer of a subframe from external memory to L2.

The same PaRAM entry options are used for QDMA channels, as well as DMA channels. The EDMA\_TPCC\_OPT\_n[3] STATIC bit is set to prevent linking. For successive transfers, only changed parameters need to be programmed before triggering the channel.

Figure 11-25 shows the parameters for Subframe Extraction transfer.



**Figure 11-24. Subframe Extraction Transfer**

**Figure 11-25. Subframe Extraction Example PaRAM Configuration**

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 000Ch		Channel Options Parameter (OPT)	
Channel Source Address (SRC)		Channel Source Address (SRC)	
000Ch	0020h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)		Channel Destination Address (DST)	
0020h	0500h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0000h	0000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA\_TPCC\_OPT\_n[2] SYNCDIM = 0x1
- EDMA\_TPCC\_OPT\_n[3] STATIC = 0x1
- EDMA\_TPCC\_OPT\_n[20] TCINTEN = 0x1

### 11.4.3 Data Sorting Example

Many applications require the use of multiple data arrays, it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion of contiguous memory spaces. For these instances, the EDMA can reorganize the data into the desired format.

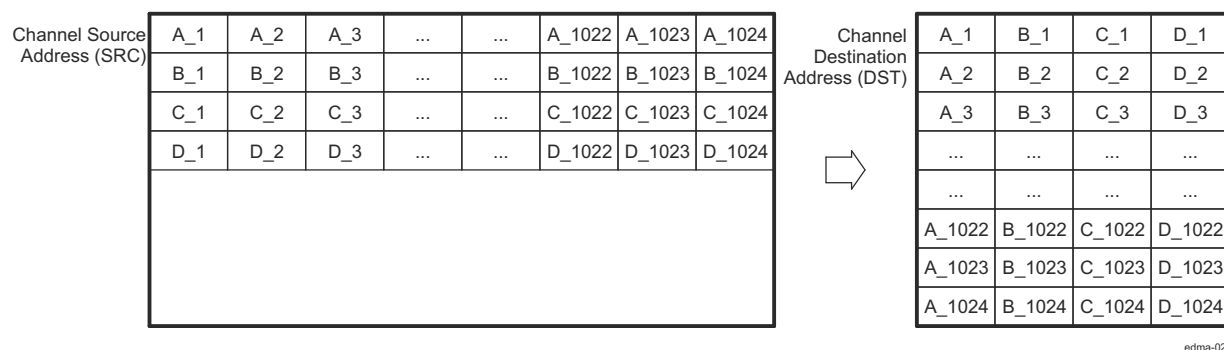
To determine the parameter set values, the following need to be considered:

- ACNT - Program this to be the size in bytes of an element.
- BCNT - Program this to be the number of elements in a frame.
- CCNT - Program this to be the number of frames.
- SBIDX - Program this to be the size of the element or ACNT.
- DBIDX - CCNT × ACNT
- SCIDX - ACNT × BCNT
- DCIDX - ACNT

The synchronization type needs to be AB-synchronized and the EDMA\_TPCC\_OPT\_n[3] STATIC bit is 0 to allow updates to the parameter set. It is advised to use normal EDMA channels for sorting.

It is not possible to sort this with a single trigger event. Instead, the channel can be programmed to be chained to itself. After BCNT elements get sorted, intermediate chaining could be used to trigger the channel again causing the transfer of the next BCNT elements and so on. [Figure 11-27](#) shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

[Figure 11-26](#) shows the Data Sorting transfer



**Figure 11-26. Data Sorting Example**

edma-023



**Figure 11-27. Data Sorting Example PaRAM Configuration**

(a) EDMA Parameters

Parameter Contents			Parameter	
0090 0004h			Channel Options Parameter (OPT)	
Channel Source Address (SRC)			Channel Source Address (SRC)	
0400h	0004h		Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)			Channel Destination Address (DST)	
0010h	0001h		Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
0000h	FFFFh		BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h		Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
0000h	0004h		Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA\_TPCC\_OPT\_n[2] SYNCDIM = 0x1
- EDMA\_TPCC\_OPT\_n[20] TCINTEN = 0x1
- EDMA\_TPCC\_OPT\_n[23] ITCCHEN = 0x1

### 11.4.4 Setting Up an EDMA Transfer

The following list provides a quick guide for the typical steps involved in setting up a transfer.

1. Initiating a DMA/QDMA channel
  - a. Determine the type of channel (QDMA or DMA) to be used.
  - b. Channel mapping
    - i. If using a QDMA channel, program the EDMA\_TPCC\_QCHMAPN\_j with the parameter set number to which the channel maps and the trigger word.
    - ii. If using a DMA channel, program the EDMA\_TPCC\_DCHMAPN\_m with the parameter set number to which the channel maps.
  - c. If the channel is being used in the context of a shadow region, ensure the EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k for the region is properly set up to allow read write accesses to bits in the event registers and interrupt registers in the Shadow region memory map. The subsequent steps in this process should be done using the respective shadow region registers. (Shadow region descriptions and usage are provided in [Section 11.3.7.1.](#))
  - d. Determine the type of triggering used.
    - i. If external events are used for triggering (DMA channels), enable the respective event in EDMA\_TPCC\_EER / EDMA\_TPCC\_EERH by writing into EDMA\_TPCC\_EESR / EDMA\_TPCC\_EESRH.
    - ii. If QDMA Channel is used, enable the channel in EDMA\_TPCC\_QEER by writing into EDMA\_TPCC\_QEESR.
  - e. Queue setup
    - i. If a QDMA channel is used, set up the EDMA\_TPCC\_QDMAQNUM to map the channel to the respective event queue.
    - ii. If a DMA channel is used, set up the EDMA\_TPCC\_DMAQNUMN\_k to map the event to the respective event queue.
2. Parameter set setup
  - a. Program the PaRAM set number associated with the channel. Note that

#### Note

If it is a QDMA channel, the PaPARAM entry that is configured as trigger word is written to last. Alternatively, enable the QDMA channel (step 1-d-ii above) just before the write to the trigger word.

3. Interrupt setup
  - a. Enable the interrupt in the EDMA\_TPCC\_IER / EDMA\_TPCC\_IERH by writing into EDMA\_TPCC\_IESR / EDMA\_TPCC\_IESRH.
  - b. Ensure the EDMA\_TPCC completion interrupt (this refers to either the Global interrupt or the shadow region interrupt) is enabled properly in the Device Interrupt controller.
  - c. Set up the interrupt controller properly to receive the expected EDMA interrupt.
4. Initiate transfer
  - a. This step is highly dependent on the event trigger source:
    - i. If the source is an external event coming from a peripheral, the peripheral will be enabled to start generating relevant EDMA events that can be latched to the EDMA\_TPCC\_ER transfer.
    - ii. For QDMA events, writes to the trigger word (step 2-a above) will initiate the transfer.
    - iii. Manually triggered transfers will be initiated by writes to the Event Set Registers EDMA\_TPCC\_ESR / EDMA\_TPCC\_ESRH.
    - iv. Chained-trigger events initiate when a previous transfer returns a transfer completion code equal to the chained channel number.
5. Wait for completion

- a. If the interrupts are enabled as mentioned in step 3 above, then the EDMA\_TPCC will generate a completion interrupt to the CPU whenever transfer completion results in setting the corresponding bits in the interrupt pending register EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH. The set bits must be cleared in the EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH by writing to corresponding bit in EDMA\_TPCC\_ICR / EDMA\_TPCC\_ICRH.
- b. If polling for completion (interrupts not enabled in the device controller), then the application code can wait on the expected bits to be set in the EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH. Again, the set bits in the EDMA\_TPCC\_IPR / EDMA\_TPCC\_IPRH must be manually cleared via EDMA\_TPCC\_ICR / EDMA\_TPCC\_ICRH before the next set of transfers is performed for the same transfer completion code values.

## 11.5 EDMA Debug Checklist and Programming Tips

This section lists some tips to keep in mind while debugging applications using the EDMA controller.

### 11.5.1 EDMA Debug Checklist

Table 11-25 provides some common issues and their probable causes and resolutions.

**Table 11-25. Debug Checklist**

Issue	Description/Solution
The transfer associated with the channel does not happen. The channel does not get serviced.	The EDMA_TPCC may not service a transfer request, even though the associated PaRAM set is programmed appropriately. Check for the following: 1) Verify that events are enabled, i.e., if an external/peripheral event is latched in Event Registers EDMA_TPCC_ER / EDMA_TPCC_ERH, check that the event is enabled in the Event Enable Registers EDMA_TPCC_EER / EDMA_TPCC_EERH. Similarly, for QDMA channels, check that QDMA events are appropriately enabled in the QDMA Event Enable Register EDMA_TPCC_QEER. 2) Verify that the DMA or QDMA Secondary Event Register EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER bits corresponding to the particular event or channel are not set.
The Secondary Event Registers bits are set, not allowing additional transfers to occur on a channel.	It is possible that a trigger event was received when the parameter set associated with the channel/event was a NULL set for a previous transfer on the channel. This is typical in two cases: 1) QDMA channels: Typically if the parameter set is non-static and expected to be terminated by a NULL set (i.e., EDMA_TPCC_OPT_n[3] STATIC = 0x0, EDMA_TPCC_LNK_n[15:0] LINK = 0xFFFF), the parameter set is updated with a NULL set after submission of the last TR. Because QDMA channels are auto-triggered, this update caused the generation of an event. An event generated for a NULL set causes an error condition and results in setting the bits corresponding to the QDMA channel in the EDMA_TPCC_QEMR and EDMA_TPCC_QSER. This will disable further prioritization of the channel. 2) DMA channels used in a continuous mode: The peripheral may be set up to continuously generate infinite events. The parameter set may be programmed to expect only a finite number of events and to be terminated by a NULL link. After the expected number of events, the parameter set is reloaded with a NULL parameter set. Because the peripheral will generate additional events, an error condition is set in the EDMA_TPCC_SER[31:0] En and EDMA_TPCC_EMR[31:0] En set, preventing further event prioritization. Check the number of events received is limited to the expected number of events for which the parameter set is programmed, or check the bits corresponding to particular channel or event are not set in the Secondary event registers (EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER) and Event Missed Registers (EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR) before trying to perform subsequent transfers for the event/channel.

**Table 11-25. Debug Checklist (continued)**

Issue	Description/Solution
Completion interrupts are not asserted, or no further interrupts are received after the first completion interrupt.	<p>Check the following:</p> <ol style="list-style-type: none"> <li>1) The interrupt generation is enabled in the EDMA_TPCC_OPT_n of the associated PaRAM set (EDMA_TPCC_OPT_n[20] TCINTEN = 0x1 and/or EDMA_TPCC_OPT_n[20] ITCINTEN = 0x1).</li> <li>2) The interrupts are enabled in the EDMA Channel Controller, via the Interrupt Enable Registers (EDMA_TPCC_IER / EDMA_TPCC_IERH ).</li> <li>3) The corresponding interrupts are enabled in the device interrupt controller.</li> <li>4) The set interrupts are cleared in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) before exiting the transfer completion interrupt service routine (ISR). See <a href="#">Section 11.3.9.1.2 Clearing Transfer Completion Interrupts</a> for details on writing EDMA ISRs.</li> <li>5) If working with shadow region interrupts, make sure that the DMA Region Access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k ) are set up properly, because the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers act as secondary enables for shadow region completion interrupts, along with the EDMA_TPCC_IER / EDMA_TPCC_IERH registers.</li> </ol> <p>If working with shadow region interrupts, make sure that the bits corresponding to the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value are also enabled in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers. For instance, if the PaRAM set associated with Channel 0 returns a completion code of 63 EDMA_TPCC_OPT_n[17:12] TCC = 63, ensure that EDMA_TPCC_DRAEHM_k[31] E63 is also set for a shadow region completion interrupt because the interrupt pending register bit set will be EDMA_TPCC_IPRH[31] I63 (not EDMA_TPCC_IPR[0] I0).</p>

## 11.5.2 EDMA Programming Tips

- For several registers, the setting and clearing of bits needs to be done via separate dedicated registers. For example, the Event Register (EDMA\_TPCC\_ER / EDMA\_TPCC\_ERH) can only be cleared by writing a 1 to the corresponding bits in the Event Clear Registers (EDMA\_TPCC\_ECR / EDMA\_TPCC\_ECRH). Similarly, the Event Enable Register (EDMA\_TPCC\_EER / EDMA\_TPCC\_EERH) bits can only be set with writing of 0x1 to the Event Enable Set Registers (EDMA\_TPCC\_EESR / EDMA\_TPCC\_EESRH) and cleared with writing of 0x1 to the corresponding bits in the Event Enable Clear Register (EDMA\_TPCC\_EECR / EDMA\_TPCC\_EECRH).
- Writes to the shadow region memory maps are governed by region access registers (EDMA\_TPCC\_DRAE / EDMA\_TPCC\_DRAEHM\_k / EDMA\_TPCC\_QRAEN\_k). If the appropriate channels are not enabled in these registers, read/write access to the shadow region memory map is not enabled.
- When working with shadow region completion interrupts, ensure that the DMA Region Access Registers (EDMA\_TPCC\_DRAEM\_k / EDMA\_TPCC\_DRAEHM\_k) for every region are set in a mutually exclusive way (unless it is a requirement for an application). If there is an overlap in the allocated channels and transfer completion codes (setting of Interrupt Pending Register bits) in the region resource allocation, it results in multiple shadow region completion interrupts.  
For example, if EDMA\_TPCC\_DRAEM\_k.DRAEM\_0[0] E0 and EDMA\_TPCC\_DRAEM\_k.DRAEM\_1[0] E0 are both set, then on completion of a transfer that returns a TCC = 0x0, they will generate both shadow region 0 and 1 completion interrupts.
- While programming a non-dummy parameter set, ensure the EDMA\_TPCC\_CCNT\_n[15:0] CCNT is not left to zero.
- Enable the EDMA\_TPCC error interrupt in the device controller and attach an interrupt service routine (ISR) to ensure that error conditions are not missed in an application and are appropriately addressed with the ISR.
- Depending on the application, it can want to break large transfers into smaller transfers and use self-chaining to prevent starvation of other events in an event queue.
- In applications where a large transfer is broken into sets of small transfers using chaining or other methods, it chooses to use the early chaining option to reduce the time between the sets of transfers and increase the throughput.  
However, keep in mind that with early completion, all data might have not been received at the end point when completion is reported because the EDMA\_TPCC internally signals completion when the TR is submitted to the EDMA\_TPTC, potentially before any data has been transferred.
- The event queue entries can be observed to determine the last few events if there is a system failure (provided the entries were not bypassed).

## 11.6 EDMA Event Map

### 11.6.1 MSS TPCC\_A Event Map

**Table 11-26. MSSTPCC\_A Event Map**

TPCC Event	DefineName	Description
0	MSS_SPIA_DMA_REQ0	MSSSPIA DMA Request 0
1	MSS_SPIA_DMA_REQ1	MSSSPIA DMA Request 1
2	MSS_SPIA_DMA_REQ2	MSSSPIA DMA Request 2
3	MSS_SPIA_DMA_REQ3	MSSSPIA DMA Request 3
4	MSS_SPIA_DMA_REQ4	MSSSPIA DMA Request 4
5	MSS_SPIA_DMA_REQ5	MSSSPIA DMA Request 5
6	MSS_SPIB_DMA_REQ0	MSSSPIB DMA Request 0
7	MSS_SPIB_DMA_REQ1	MSSSPIB DMA Request 1
8	MSS_SPIB_DMA_REQ2	MSSSPIB DMA Request 2
9	MSS_SPIB_DMA_REQ3	MSSSPIB DMA Request 3
10	MSS_SPIB_DMA_REQ4	MSSSPIB DMA Request 4

**Table 11-26. MSSTPCC\_A Event Map (continued)**

TPCC Event	DefineName	Description
11	MSS_SPIB_DMA_REQ5	MSSSPIB DMA Request 5
12	MSS_QSPI_DMA_REQ0	MSS_QSPI_DMA_REQ0
13	MSS_MCRC_DMA_REQ0	MSSMCRC DMA Request 0
14	MSS_MCRC_DMA_REQ1	MSSMCRC DMA Request 1
15	MSS_RTIA_DMA_REQ0	MSSRTIA DMA Request 0
16	MSS_RTIA_DMA_REQ1	MSSRTIA DMA Request 1
17	MSS_RTIA_DMA_REQ2	MSSRTIA DMA Request 2
18	MSS_RTIA_DMA_REQ3	MSSRTIA DMA Request 3
19	MSS_RTIB_DMA_REQ0	MSSRTIB DMA Request 0
20	MSS_RTIB_DMA_REQ1	MSSRTIB DMA Request 1
21	MSS_RTIC_DMA_REQ0	MSSRTIC DMA Request 0
22	MSS_RTIC_DMA_REQ1	MSSRTIC DMA Request 1
23	MSS_WDT_DMA_REQ0	MSSWDT DMA Request 0
24	MSS_WDT_DMA_REQ1	MSSWDT DMA Request 1
25	MSS_WDT_DMA_REQ2	MSSWDT DMA Request 2
26	MSS_WDT_DMA_REQ3	MSSWDT DMA Request 3
27	MSS_ETPWMA_DMA_REQ0	MSS_ETPWMADMA Req directly taken from IO
28	MSS_ETPWMA_DMA_REQ1	MSS_ETPWMADMA Req directly taken from IO
29	MSS_ETPWMB_DMA_REQ0	MSS_ETPWMBDMA Req directly taken from IO
30	MSS_ETPWMB_DMA_REQ1	MSS_ETPWMBDMA Req directly taken from IO
31	MSS_ETPWMCDMA_REQ0	MSS_ETPWMCDMA Req directly taken from IO
32	MSS_ETPWMCDMA_REQ1	MSS_ETPWMCDMA Req directly taken from IO
33	MSS_MCANA_DMA_REQ0	MSS_MCANADMA Request 0
34	MSS_MCANA_DMA_REQ1	MSS_MCANADMA Request 1
35	MSS_MCANA_FE_INT1	MSS_MCANAfilter event 1
36	MSS_MCANA_FE_INT2	MSS_MCANAfilter event 2
37	MSS_MCANA_FE_INT4	MSS_MCANAfilter event 4
38	MSS_MCANB_DMA_REQ0	MSS_MCANBDMA Request 0
39	MSS_MCANB_DMA_REQ1	MSS_MCANBDMA Request 1
40	MSS_MCANB_FE_INT1	MSS_MCANBfilter event 1
41	MSS_MCANB_FE_INT2	MSS_MCANBfilter event 2
42	MSS_MCANB_FE_INT4	MSS_MCANBfilter event 4
43	MSS_RTIB_DMA_REQ2	MSSRTIB DMA Request 2
44	MSS_RTIB_DMA_REQ3	MSSRTIB DMA Request 3
45	MSS_RTIC_DMA_REQ2	MSSRTIC DMA Request 2
46	MSS_RTIC_DMA_REQ3	MSSRTIC DMA Request 3
47	RESERVED	RESERVED
48	RESERVED	RESERVED
49	MSS_GIO_PAD_INT0	InterruptTriger from GIO[0][0]
50	MSS_GIO_PAD_INT1	InterruptTriger from GIO[0][1]
51	MSS_GIO_PAD_INT2	InterruptTriger from GIO[0][2]
52	MSS_GIO_PAD_INT3	InterruptTriger from GIO[0][3]
53	MSS_GIO_PAD_INT4	InterruptTriger from GIO[1][0]
54	RESERVED	RESERVED
55	MSS_I2C_DMA_REQ 0	MSS_I2CDMA Request 0

**Table 11-26. MSSTPCC\_A Event Map (continued)**

TPCC Event	DefineName	Description
56	MSS_I2C_DMA_REQ 1	MSS_I2CDMA Request 1
57	MSS_SCIA_RX_DMA_REQ	MSSSCIA RX DMA Request
58	MSS_SCIA_TX_DMA_REQ	MSSSCIA TX DMA Request
59	MSS_SCIB_RX_DMA_REQ	MSSSCIB RX DMA Request
60	MSS_SCIB_TX_DMA_REQ	MSSSCIB TX DMA Request
61	RESERVED	RESERVED
62	RESERVED	RESERVED
63	CBUFF_DMA_REQ	Triggerfrom CBUFF

### 11.6.2 MSS TPCC\_B Event Map

**Table 11-27. MSS TPCC\_B Event Map**

TPCC Event	Define Name	Description
0	MSS_MCRC_DMA_REQ0	MSS MCRC DMA Request 0
1	MSS_MCRC_DMA_REQ1	MSS MCRC DMA Request 1
2	MSS_ETPWMA_DMA_REQ0	MSS_ETPWMA DMA Req directly taken from IO
3	MSS_ETPWMA_DMA_REQ1	MSS_ETPWMA DMA Req directly taken from IO
4	MSS_ETPWMB_DMA_REQ0	MSS_ETPWMB DMA Req directly taken from IO
5	MSS_ETPWMB_DMA_REQ1	MSS_ETPWMB DMA Req directly taken from IO
6	MSS_ETPWMC_DMA_REQ0	MSS_ETPWMC DMA Req directly taken from IO
7	MSS_ETPWMC_DMA_REQ1	MSS_ETPWMC DMA Req directly taken from IO
8	MSS_MCANA_DMA_REQ0	MSS_MCANA DMA Request 0
9	MSS_MCANA_DMA_REQ1	MSS_MCANA DMA Request 1
10	MSS_MCANB_DMA_REQ0	MSS_MCANB DMA Request 0
11	MSS_MCANB_DMA_REQ1	MSS_MCANB DMA Request 1
12	Reserved	Reserved
13	Reserved	Reserved
14	Reserved	Reserved
15	Reserved	Reserved
16	MSS_GIO_PAD_INT0	Interrupt Triger from GIO[0][0]
17	MSS_GIO_PAD_INT1	Interrupt Triger from GIO[0][1]
18	MSS_GIO_PAD_INT2	Interrupt Triger from GIO[0][2]
19	MSS_GIO_PAD_INT3	Interrupt Triger from GIO[0][3]
20	MSS_GIO_PAD_INT4	Interrupt Triger from GIO[1][0]
21	Reserved	Reserved
22	Reserved	Reserved
23	Reserved	Reserved
24	Reserved	Reserved
25	Reserved	Reserved
26	DTHE_SHA_DMA_REQ0	DMA request 1 from SHA
27	DTHE_SHA_DMA_REQ1	DMA request 2 from SHA
28	DTHE_SHA_DMA_REQ2	DMA request 3 from SHA
29	DTHE_SHA_DMA_REQ3	DMA request 4 from SHA
30	DTHE_SHA_DMA_REQ4	DMA request 5 from SHA
31	DTHE_SHA_DMA_REQ5	DMA request 6 from SHA



**Table 11-27. MSS TPCC\_B Event Map (continued)**

TPCC Event	Define Name	Description
32	DTHE_AES_DMA_REQ0	DMA request 1 from AES
33	DTHE_AES_DMA_REQ1	DMA request 2 from AES
34	DTHE_AES_DMA_REQ2	DMA request 3 from AES
35	DTHE_AES_DMA_REQ3	DMA request 4 from AES
36	DTHE_AES_DMA_REQ4	DMA request 5 from AES
37	DTHE_AES_DMA_REQ5	DMA request 6 from AES
38	DTHE_AES_DMA_REQ6	DMA request 7 from AES
39	DTHE_AES_DMA_REQ7	DMA request 8 from AES
40	Reserved	Reserved
41	Reserved	Reserved
42	Reserved	Reserved
43	MSS_MCANB_FE_INT1	Filter event 1 from MSS_MCANA
44	MSS_MCANB_FE_INT2	Filter event 2 from MSS_MCANA
45	MSS_MCANB_FE_INT3	Filter event 3 from MSS_MCANA
46	MSS_MCANB_FE_INT4	Filter event 4 from MSS_MCANA
47	MSS_MCANB_FE_INT5	Filter event 5 from MSS_MCANA
48	MSS_MCANB_FE_INT6	Filter event 6 from MSS_MCANA
49	MSS_MCANB_FE_INT7	Filter event 7 from MSS_MCANA
50	MSS_MCANB_FE_INT1	Filter event 1 from MSS_MCANB
51	MSS_MCANB_FE_INT2	Filter event 2 from MSS_MCANB
52	MSS_MCANB_FE_INT3	Filter event 3 from MSS_MCANB
53	MSS_MCANB_FE_INT4	Filter event 4 from MSS_MCANB
54	MSS_MCANB_FE_INT5	Filter event 5 from MSS_MCANB
55	MSS_MCANB_FE_INT6	Filter event 6 from MSS_MCANB
56	MSS_MCANB_FE_INT7	Filter event 7 from MSS_MCANB
57	Reserved	Reserved
58	Reserved	Reserved
59	Reserved	Reserved
60	Reserved	Reserved
61	Reserved	Reserved
62	Reserved	Reserved
63	Reserved	Reserved
57	DTHE_SM4_DMA_REQ0	DMA request 1 from SM4
58	DTHE_SM4_DMA_REQ1	DMA request 2 from SM4
59	DTHE_SM4_DMA_REQ2	DMA request 3 from SM4
60	DTHE_SM4_DMA_REQ3	DMA request 4 from SM4
61	DTHE_SM3_DMA_REQ0	DMA request 1 from SM3
62	DTHE_SM3_DMA_REQ1	DMA request 2 from SM3
63	DTHE_SM3_DMA_REQ2	DMA request 3 from SM3

### 11.6.3 DSS TPCC\_A Event Map

**Table 11-28. DSS TPCC\_A Event Map**

TPCC Event	Define Name	Description
0	DSS_RTIA_DMA_REQ0	DSS RTIA DMA Request 0
1	DSS_RTIA_DMA_REQ1	DSS RTIA DMA Request 1
2	DSS_RTIA_DMA_REQ2	DSS RTIA DMA Request 2
3	DSS_RTIA_DMA_REQ3	DSS RTIA DMA Request 3
4	DSS_RTIB_DMA_REQ0	DSS RTIB DMA Request 0
5	DSS_RTIB_DMA_REQ1	DSS RTIB DMA Request 1
6	DSS_RTIB_DMA_REQ2	DSS RTIB DMA Request 2
7	DSS_RTIB_DMA_REQ3	DSS RTIB DMA Request 3
8	DSS_WDT_DMA_REQ0	DSS WDT DMA Request 0
9	DSS_WDT_DMA_REQ1	DSS WDT DMA Request 1
10	DSS_WDT_DMA_REQ2	DSS WDT DMA Request 2
11	DSS_WDT_DMA_REQ3	DSS WDT DMA Request 3
12	DSS_MCRC_DMA_REQ0	DSS MCRC DMA Request 0
13	DSS_MCRC_DMA_REQ1	DSS MCRC DMA Request 1
14	DSS_SCIA_RX_DMA_REQ	DSS SCIA RX DMA Request
15	DSS_SCIA_TX_DMA_REQ	DSS SCIA TX DMA Request
16	RESERVED	RESERVED
17	RESERVED	RESERVED
18	DSS_CBUFF_DMA_REQ0	DSS CBUF DMA Request 0
19	DSS_CBUFF_DMA_REQ1	DSS CBUF DMA Request 1
20	DSS_CBUFF_DMA_REQ2	DSS CBUF DMA Request 2
21	DSS_CBUFF_DMA_REQ3	DSS CBUF DMA Request 3
22	DSS_CBUFF_DMA_REQ4	DSS CBUF DMA Request 4
23	DSS_CBUFF_DMA_REQ5	DSS CBUF DMA Request 5
24	DSS_CBUFF_DMA_REQ6	DSS CBUF DMA Request 6
25	ADC_VALID_FALL_EDGE	Interrupt is trigger during Falling edge of ADC valid
26	DFE_FRAME_START_TO_MSS	Frame start interrupt from BSS which is masked with "BSS_GPCFG::MSS_FS_INTR_MASK" register
27	DFE_CHIRP_CYCLE_START	Chirp cycle start interrupt from dfe
28	DFE_CHIRP_CYCLE_END	Chirp cycle end interrupt from dfe
29	DFE_END_OF_FRAME	End of Frame interrupt from Dfe
30	RSS_FRC_FRAME_START	Frame start interrupt from FRC
31	RSS_ADC_CAPTURE_COMPLETE	ADC capture complete from DFE-DSP bridge
32	DSS_HWA_DMA_REQ0	DSS HWA DMA Request 0
33	DSS_HWA_DMA_REQ1	DSS HWA DMA Request 1
34	DSS_HWA_DMA_REQ2	DSS HWA DMA Request 2
35	DSS_HWA_DMA_REQ3	DSS HWA DMA Request 3
36	DSS_HWA_DMA_REQ4	DSS HWA DMA Request 4
37	DSS_HWA_DMA_REQ5	DSS HWA DMA Request 5
38	DSS_HWA_DMA_REQ6	DSS HWA DMA Request 6
39	DSS_HWA_DMA_REQ7	DSS HWA DMA Request 7
40	DSS_HWA_DMA_REQ8	DSS HWA DMA Request 8

**Table 11-28. DSS TPCC\_A Event Map (continued)**

TPCC Event	Define Name	Description
41	DSS_HWA_DMA_REQ9	DSS HWA DMA Request 9
42	DSS_HWA_DMA_REQ10	DSS HWA DMA Request 10
43	DSS_HWA_DMA_REQ11	DSS HWA DMA Request 11
44	DSS_HWA_DMA_REQ12	DSS HWA DMA Request 12
45	DSS_HWA_DMA_REQ13	DSS HWA DMA Request 13
46	DSS_HWA_DMA_REQ14	DSS HWA DMA Request 14
47	DSS_HWA_DMA_REQ15	DSS HWA DMA Request 15
48	DSS_HWA_DMA_REQ16	DSS HWA DMA Request 16
49	DSS_HWA_DMA_REQ17	DSS HWA DMA Request 17
50	DSS_HWA_DMA_REQ18	DSS HWA DMA Request 18
51	DSS_HWA_DMA_REQ19	DSS HWA DMA Request 19
52	DSS_HWA_DMA_REQ20	DSS HWA DMA Request 20
53	DSS_HWA_DMA_REQ21	DSS HWA DMA Request 21
54	DSS_HWA_DMA_REQ22	DSS HWA DMA Request 22
55	DSS_HWA_DMA_REQ23	DSS HWA DMA Request 23
56	DSS_HWA_DMA_REQ24	DSS HWA DMA Request 24
57	DSS_HWA_DMA_REQ25	DSS HWA DMA Request 25
58	DSS_HWA_DMA_REQ26	DSS HWA DMA Request 26
59	DSS_HWA_DMA_REQ27	DSS HWA DMA Request 27
60	DSS_HWA_DMA_REQ28	DSS HWA DMA Request 28
61	DSS_HWA_DMA_REQ29	DSS HWA DMA Request 29
62	DSS_HWA_DMA_REQ30	DSS HWA DMA Request 30
63	DSS_HWA_DMA_REQ31	DSS HWA DMA Request 31

#### 11.6.4 DSS TPCC\_B / TPCC\_C Event Map

**Table 11-29. DSS TPCC\_B Event Map**

TPCC Event	Define Name	Description
0	DSS_RTIA_DMA_REQ0	DSS RTIA DMA Request 0
1	DSS_RTIA_DMA_REQ1	DSS RTIA DMA Request 1
2	DSS_RTIA_DMA_REQ2	DSS RTIA DMA Request 2
3	DSS_RTIA_DMA_REQ3	DSS RTIA DMA Request 3
4	DSS_RTIB_DMA_REQ0	DSS RTIB DMA Request 0
5	DSS_RTIB_DMA_REQ1	DSS RTIB DMA Request 1
6	DSS_RTIB_DMA_REQ2	DSS RTIB DMA Request 2
7	DSS_RTIB_DMA_REQ3	DSS RTIB DMA Request 3
8	DSS_WDT_DMA_REQ0	DSS WDT DMA Request 0
9	DSS_WDT_DMA_REQ1	DSS WDT DMA Request 1
10	DSS_WDT_DMA_REQ2	DSS WDT DMA Request 2
11	DSS_WDT_DMA_REQ3	DSS WDT DMA Request 3
12	DSS_MCRC_DMA_REQ0	DSS MCRC DMA Request 0
13	DSS_MCRC_DMA_REQ1	DSS MCRC DMA Request 1
14	DSS_SCIA_RX_DMA_REQ	DSS SCIA RX DMA Request
15	DSS_SCIA_TX_DMA_REQ	DSS SCIA TX DMA Request
16	RCSS_CSI2A_EOF_INT	RCSS CSI2A End of Frame Interrupt (all contexts combined interrupt)

**Table 11-29. DSS TPCC\_B Event Map (continued)**

TPCC Event	Define Name	Description
17	RCSS_CSI2A_EOL_INT	RCSS CSI2A End of Line Interrupt (all contexts combined interrupt)
18	DSS_CBUFF_DMA_REQ0	DSS CBUF DMA Request 0
19	DSS_CBUFF_DMA_REQ1	DSS CBUF DMA Request 1
20	DSS_CBUFF_DMA_REQ2	DSS CBUF DMA Request 2
21	DSS_CBUFF_DMA_REQ3	DSS CBUF DMA Request 3
22	DSS_CBUFF_DMA_REQ4	DSS CBUF DMA Request 4
23	DSS_CBUFF_DMA_REQ5	DSS CBUF DMA Request 5
24	DSS_CBUFF_DMA_REQ6	DSS CBUF DMA Request 6
25	RCSS_CSI2A_SOF_INT0	RCSS CSI2A Start of Frame Interrupt 0(Selective frame start based on Register RCSS_CSI2A_CFG in RCSS_CTRL)
26	RCSS_CSI2A_SOF_INT1	RCSS CSI2A Start of Frame Interrupt 1(Selective frame start based on Register RCSS_CSI2A_CFG in RCSS_CTRL)
27	RCSS_CSI2A_EOL_CNTX0_INT	RCSS_CSI2A End of Line Interrupt for Context 0
28	RCSS_CSI2A_EOL_CNTX1_INT	RCSS_CSI2A End of Line Interrupt for Context 1
29	RCSS_CSI2A_EOL_CNTX2_INT	RCSS_CSI2A End of Line Interrupt for Context 2
30	RCSS_CSI2A_EOL_CNTX3_INT	RCSS_CSI2A End of Line Interrupt for Context 3
31	Reserved	Reserved
32	DSS_HWA_DMA_REQ0	DSS HWA DMA Request 0
33	DSS_HWA_DMA_REQ1	DSS HWA DMA Request 1
34	DSS_HWA_DMA_REQ2	DSS HWA DMA Request 2
35	DSS_HWA_DMA_REQ3	DSS HWA DMA Request 3
36	DSS_HWA_DMA_REQ4	DSS HWA DMA Request 4
37	DSS_HWA_DMA_REQ5	DSS HWA DMA Request 5
38	DSS_HWA_DMA_REQ6	DSS HWA DMA Request 6
39	DSS_HWA_DMA_REQ7	DSS HWA DMA Request 7
40	DSS_HWA_DMA_REQ8	DSS HWA DMA Request 8
41	DSS_HWA_DMA_REQ9	DSS HWA DMA Request 9
42	DSS_HWA_DMA_REQ10	DSS HWA DMA Request 10
43	DSS_HWA_DMA_REQ11	DSS HWA DMA Request 11
44	DSS_HWA_DMA_REQ12	DSS HWA DMA Request 12
45	DSS_HWA_DMA_REQ13	DSS HWA DMA Request 13
46	DSS_HWA_DMA_REQ14	DSS HWA DMA Request 14
47	DSS_HWA_DMA_REQ15	DSS HWA DMA Request 15
48	DSS_HWA_DMA_REQ16	DSS HWA DMA Request 16
49	DSS_HWA_DMA_REQ17	DSS HWA DMA Request 17
50	DSS_HWA_DMA_REQ18	DSS HWA DMA Request 18
51	DSS_HWA_DMA_REQ19	DSS HWA DMA Request 19
52	DSS_HWA_DMA_REQ20	DSS HWA DMA Request 20
53	DSS_HWA_DMA_REQ21	DSS HWA DMA Request 21
54	DSS_HWA_DMA_REQ22	DSS HWA DMA Request 22
55	DSS_HWA_DMA_REQ23	DSS HWA DMA Request 23

**Table 11-29. DSS TPCC\_B Event Map (continued)**

TPCC Event	Define Name	Description
56	DSS_HWA_DMA_REQ24	DSS HWA DMA Request 24
57	DSS_HWA_DMA_REQ25	DSS HWA DMA Request 25
58	DSS_HWA_DMA_REQ26	DSS HWA DMA Request 26
59	DSS_HWA_DMA_REQ27	DSS HWA DMA Request 27
60	DSS_HWA_DMA_REQ28	DSS HWA DMA Request 28
61	DSS_HWA_DMA_REQ29	DSS HWA DMA Request 29
62	DSS_HWA_DMA_REQ30	DSS HWA DMA Request 30
63	DSS_HWA_DMA_REQ31	DSS HWA DMA Request 31

**Table 11-30. DSS TPCC\_C Event Map**

TPCC Event	Define Name	Description
0	DSS_RTIA_DMA_REQ0	DSS RTIA DMA Request 0
1	DSS_RTIA_DMA_REQ1	DSS RTIA DMA Request 1
2	DSS_RTIA_DMA_REQ2	DSS RTIA DMA Request 2
3	DSS_RTIA_DMA_REQ3	DSS RTIA DMA Request 3
4	DSS_RTIB_DMA_REQ0	DSS RTIB DMA Request 0
5	DSS_RTIB_DMA_REQ1	DSS RTIB DMA Request 1
6	DSS_RTIB_DMA_REQ2	DSS RTIB DMA Request 2
7	DSS_RTIB_DMA_REQ3	DSS RTIB DMA Request 3
8	DSS_WDT_DMA_REQ0	DSS WDT DMA Request 0
9	DSS_WDT_DMA_REQ1	DSS WDT DMA Request 1
10	DSS_WDT_DMA_REQ2	DSS WDT DMA Request 2
11	DSS_WDT_DMA_REQ3	DSS WDT DMA Request 3
12	DSS_MCRC_DMA_REQ0	DSS MCRC DMA Request 0
13	DSS_MCRC_DMA_REQ1	DSS MCRC DMA Request 1
14	DSS_SCIA_RX_DMA_REQ	DSS SCIA RX DMA Request
15	DSS_SCIA_TX_DMA_REQ	DSS SCIA TX DMA Request
16	RSS_ADC_CAPTURE_COMPLETE_DITH	ADC capture complete interrupt from DFE-DSP bridge after dithering
17	Reserved	Reserved
18	DSS_CBUFF_DMA_REQ0	DSS CBUF DMA Request 0
19	DSS_CBUFF_DMA_REQ1	DSS CBUF DMA Request 1
20	DSS_CBUFF_DMA_REQ2	DSS CBUF DMA Request 2
21	DSS_CBUFF_DMA_REQ3	DSS CBUF DMA Request 3
22	DSS_CBUFF_DMA_REQ4	DSS CBUF DMA Request 4
23	DSS_CBUFF_DMA_REQ5	DSS CBUF DMA Request 5
24	DSS_CBUFF_DMA_REQ6	DSS CBUF DMA Request 6
25	Reserved	Reserved
26	Reserved	Reserved
27	Reserved	Reserved
28	Reserved	Reserved
29	Reserved	Reserved
30	Reserved	Reserved
31	Reserved	Reserved
32	DSS_HWA_DMA_REQ0	DSS HWA DMA Request 0
33	DSS_HWA_DMA_REQ1	DSS HWA DMA Request 1

**Table 11-30. DSS TPCC\_C Event Map (continued)**

TPCC Event	Define Name	Description
34	DSS_HWA_DMA_REQ2	DSS HWA DMA Request 2
35	DSS_HWA_DMA_REQ3	DSS HWA DMA Request 3
36	DSS_HWA_DMA_REQ4	DSS HWA DMA Request 4
37	DSS_HWA_DMA_REQ5	DSS HWA DMA Request 5
38	DSS_HWA_DMA_REQ6	DSS HWA DMA Request 6
39	DSS_HWA_DMA_REQ7	DSS HWA DMA Request 7
40	DSS_HWA_DMA_REQ8	DSS HWA DMA Request 8
41	DSS_HWA_DMA_REQ9	DSS HWA DMA Request 9
42	DSS_HWA_DMA_REQ10	DSS HWA DMA Request 10
43	DSS_HWA_DMA_REQ11	DSS HWA DMA Request 11
44	DSS_HWA_DMA_REQ12	DSS HWA DMA Request 12
45	DSS_HWA_DMA_REQ13	DSS HWA DMA Request 13
46	DSS_HWA_DMA_REQ14	DSS HWA DMA Request 14
47	DSS_HWA_DMA_REQ15	DSS HWA DMA Request 15
48	DSS_HWA_DMA_REQ16	DSS HWA DMA Request 16
49	DSS_HWA_DMA_REQ17	DSS HWA DMA Request 17
50	DSS_HWA_DMA_REQ18	DSS HWA DMA Request 18
51	DSS_HWA_DMA_REQ19	DSS HWA DMA Request 19
52	DSS_HWA_DMA_REQ20	DSS HWA DMA Request 20
53	DSS_HWA_DMA_REQ21	DSS HWA DMA Request 21
54	DSS_HWA_DMA_REQ22	DSS HWA DMA Request 22
55	DSS_HWA_DMA_REQ23	DSS HWA DMA Request 23
56	DSS_HWA_DMA_REQ24	DSS HWA DMA Request 24
57	DSS_HWA_DMA_REQ25	DSS HWA DMA Request 25
58	DSS_HWA_DMA_REQ26	DSS HWA DMA Request 26
59	DSS_HWA_DMA_REQ27	DSS HWA DMA Request 27
60	DSS_HWA_DMA_REQ28	DSS HWA DMA Request 28
61	DSS_HWA_DMA_REQ29	DSS HWA DMA Request 29
62	DSS_HWA_DMA_REQ30	DSS HWA DMA Request 30
63	DSS_HWA_DMA_REQ31	DSS HWA DMA Request 31

### 11.6.5 Radar Subsystem (RSS TPCC\_A) EDMA Event Map

**Table 11-31. Radar Subsystem EDMA Event Map**

TPCC Event	Define Name	Description
0	RSS_ADC_VALID	rss_int_map[8] = bss_to_mss_irq[0] - adc_valid_fall
1	RSS_FRAME_START	rss_int_map[9] = bss_to_mss_irq[1] - frame_start
2	RSS_CHIRP_CYCLE_START	rss_int_map[10] = bss_to_mss_irq[2] - chirp_cycle_start
3	RSS_CHIRP_CYCLE_END	rss_int_map[11] = bss_to_mss_irq[3] - chirp_cycle_end
4	RSS_END_OF_FRAME	rss_int_map[12] = bss_to_mss_irq[4] - end_of_frame
5	RSS_FRC_FRAME_START	rss_int_map[13]
6	RSS_ADC_CAPTURE_COMPLETE	rss_int_map[14]

**Table 11-31. Radar Subsystem EDMA Event Map (continued)**

TPCC Event	Define Name	Description
7	RSS_FRC_FRAME_END	rss_dma_map[7]
8	RSS_ADC_CAPTURE_COMPLETE	rcss_int_map[15]
9	RESERVED	RESERVED
10	RESERVED	RESERVED
11	RESERVED	RESERVED
12	RESERVED	RESERVED
13	RESERVED	RESERVED
14	RESERVED	RESERVED
15	RESERVED	RESERVED
16	RCSS_CSI2A_EOF_INT	RCSS CSI2A End Of Frame Interrupt (all contexts combined interrupt)
17	RCSS_CSI2A_EOL_INT	RCSS CSI2A End of Line Interrupt (all contexts combined interrupt)
18	RCSS_CSI2A_EOL_CNTX0_INT	RCSS_CSI2A End of Line Interrupt for Context 0
19	RCSS_CSI2A_EOL_CNTX1_INT	RCSS_CSI2A End of Line Interrupt for Context 1
20	RCSS_CSI2A_EOL_CNTX2_INT	RCSS_CSI2A End of Line Interrupt for Context 2
21	RCSS_CSI2A_EOL_CNTX3_INT	RCSS_CSI2A End of Line Interrupt for Context 3
22	RCSS_CSI2A_EOL_CNTX4_INT	RCSS_CSI2A End of Line Interrupt for Context 4
23	RCSS_CSI2A_EOL_CNTX5_INT	RCSS_CSI2A End of Line Interrupt for Context 5
24	RCSS_CSI2A_EOL_CNTX6_INT	RCSS_CSI2A End of Line Interrupt for Context 6
25	RCSS_CSI2A_EOL_CNTX7_INT	RCSS_CSI2A End of Line Interrupt for Context 7
26	RCSS_CSI2A_SOF_INT0	RCSS CSI2A Start of Frame Interrupt 0(Selective frame start based on Register RCSS_CSI2A_CFG in RCSS_CTRL)
27	RCSS_CSI2A_SOF_INT1	RCSS CSI2A Start of Frame Interrupt 1(Selective frame start based on Register RCSS_CSI2A_CFG in RCSS_CTRL)
28	RESERVED	RESERVED
29	RESERVED	RESERVED
30	DSS_CBUFF_DMA_REQ0	DSS CBUF DMA Request 0
31	DSS_CBUFF_DMA_REQ1	DSS CBUF DMA Request 1
32	DSS_CBUFF_DMA_REQ2	DSS CBUF DMA Request 2
33	DSS_CBUFF_DMA_REQ3	DSS CBUF DMA Request 3
34	DSS_CBUFF_DMA_REQ4	DSS CBUF DMA Request 4
35	DSS_CBUFF_DMA_REQ5	DSS CBUF DMA Request 5
36	DSS_CBUFF_DMA_REQ6	DSS CBUF DMA Request 6
37	RESERVED	RESERVED
38	RESERVED	RESERVED
39	RESERVED	RESERVED
40	RESERVED	RESERVED
41	RESERVED	RESERVED

**Table 11-31. Radar Subsystem EDMA Event Map (continued)**

TPCC Event	Define Name	Description
42	RESERVED	RESERVED
43	RESERVED	RESERVED
44	RESERVED	RESERVED
45	RESERVED	RESERVED
46	RESERVED	RESERVED
47	RESERVED	RESERVED
48	RESERVED	RESERVED
49	RESERVED	RESERVED
50	RESERVED	RESERVED
51	RESERVED	RESERVED
52	RESERVED	RESERVED
53	RESERVED	RESERVED
54	RESERVED	RESERVED
55	RESERVED	RESERVED
56	RESERVED	RESERVED
57	RESERVED	RESERVED
58	RESERVED	RESERVED
59	RESERVED	RESERVED
60	RESERVED	RESERVED
61	RESERVED	RESERVED
62	RESERVED	RESERVED
63	RESERVED	RESERVED

## 11.7 EDMA Request Map

For the EDMA request map, refer to [Section 11.6](#).



## **11.8 EDMA Register Manual**

### **11.8.1 EDMA Registers**

### 11.8.1.1 TPCC Registers

Table 11-32 lists the TPCC registers. All register offset addresses not listed in Table 11-32 should be considered as reserved locations and the register contents should not be modified.

**Table 11-32. TPCC Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID	<a href="#">Section 11.8.1.1.1</a>
4h	CCCFG	CCCFG	<a href="#">Section 11.8.1.1.2</a>
200h	QCHMAPN	QCHMAPN	<a href="#">Section 11.8.1.1.3</a>
240h	DMAQNUMN	DMAQNUMN	<a href="#">Section 11.8.1.1.4</a>
260h	QDMAQNUM	QDMAQNUM	<a href="#">Section 11.8.1.1.5</a>
280h	QUETCMAP	QUETCMAP	<a href="#">Section 11.8.1.1.6</a>
284h	QUEPRI	QUEPRI	<a href="#">Section 11.8.1.1.7</a>
300h	EMR	EMR	<a href="#">Section 11.8.1.1.8</a>
304h	EMRH	EMRH	<a href="#">Section 11.8.1.1.9</a>
308h	EMCR	EMCR	<a href="#">Section 11.8.1.1.10</a>
30Ch	EMCRH	EMCRH	<a href="#">Section 11.8.1.1.11</a>
310h	QEMR	QEMR	<a href="#">Section 11.8.1.1.12</a>
314h	QEMCR	QEMCR	<a href="#">Section 11.8.1.1.13</a>
318h	CCERR	CCERR	<a href="#">Section 11.8.1.1.14</a>
31Ch	CCERRCLR	CCERRCLR	<a href="#">Section 11.8.1.1.15</a>
320h	EEVAL	EEVAL	<a href="#">Section 11.8.1.1.16</a>
340h	DRAEM	DRAEM	<a href="#">Section 11.8.1.1.17</a>
344h	DRAEHM	DRAEHM	<a href="#">Section 11.8.1.1.18</a>
380h	QRAEN	QRAEN	<a href="#">Section 11.8.1.1.19</a>
400h	QNE0	QNE0	<a href="#">Section 11.8.1.1.20</a>
404h	QNE1	QNE1	<a href="#">Section 11.8.1.1.21</a>
408h	QNE2	QNE2	<a href="#">Section 11.8.1.1.22</a>
40Ch	QNE3	QNE3	<a href="#">Section 11.8.1.1.23</a>
410h	QNE4	QNE4	<a href="#">Section 11.8.1.1.24</a>
414h	QNE5	QNE5	<a href="#">Section 11.8.1.1.25</a>
418h	QNE6	QNE6	<a href="#">Section 11.8.1.1.26</a>
41Ch	QNE7	QNE7	<a href="#">Section 11.8.1.1.27</a>
420h	QNE8	QNE8	<a href="#">Section 11.8.1.1.28</a>
424h	QNE9	QNE9	<a href="#">Section 11.8.1.1.29</a>
428h	QNE10	QNE10	<a href="#">Section 11.8.1.1.30</a>
42Ch	QNE11	QNE11	<a href="#">Section 11.8.1.1.31</a>
430h	QNE12	QNE12	<a href="#">Section 11.8.1.1.32</a>
434h	QNE13	QNE13	<a href="#">Section 11.8.1.1.33</a>
438h	QNE14	QNE14	<a href="#">Section 11.8.1.1.34</a>
43Ch	QNE15	QNE15	<a href="#">Section 11.8.1.1.35</a>
600h	QSTATN	QSTATN	<a href="#">Section 11.8.1.1.36</a>
620h	QWMTHRA	QWMTHRA	<a href="#">Section 11.8.1.1.37</a>
640h	CCSTAT	CCSTAT	<a href="#">Section 11.8.1.1.38</a>
700h	AETCTL	AETCTL	<a href="#">Section 11.8.1.1.39</a>
704h	AETSTAT	AETSTAT	<a href="#">Section 11.8.1.1.40</a>
708h	AETCMD	AETCMD	<a href="#">Section 11.8.1.1.41</a>

**Table 11-32. TPCC Registers (continued)**

Offset	Acronym	Register Name	Section
1000h	ER	ER	<a href="#">Section 11.8.1.1.42</a>
1004h	ERH	ERH	<a href="#">Section 11.8.1.1.43</a>
1008h	ECR	ECR	<a href="#">Section 11.8.1.1.44</a>
100Ch	ECRH	ECRH	<a href="#">Section 11.8.1.1.45</a>
1010h	ESR	ESR	<a href="#">Section 11.8.1.1.46</a>
1014h	ESRH	ESRH	<a href="#">Section 11.8.1.1.47</a>
1018h	CER	CER	<a href="#">Section 11.8.1.1.48</a>
101Ch	CERH	CERH	<a href="#">Section 11.8.1.1.49</a>
1020h	EER	EER	<a href="#">Section 11.8.1.1.50</a>
1024h	EERH	EERH	<a href="#">Section 11.8.1.1.51</a>
1028h	EECR	EECR	<a href="#">Section 11.8.1.1.52</a>
102Ch	EECRH	EECRH	<a href="#">Section 11.8.1.1.53</a>
1030h	EESR	EESR	<a href="#">Section 11.8.1.1.54</a>
1034h	EESRH	EESRH	<a href="#">Section 11.8.1.1.55</a>
1038h	SER	SER	<a href="#">Section 11.8.1.1.56</a>
103Ch	SERH	SERH	<a href="#">Section 11.8.1.1.57</a>
1040h	SECR	SECR	<a href="#">Section 11.8.1.1.58</a>
1044h	SECRH	SECRH	<a href="#">Section 11.8.1.1.59</a>
1050h	IER	IER	<a href="#">Section 11.8.1.1.60</a>
1054h	IERH	IERH	<a href="#">Section 11.8.1.1.61</a>
1058h	IECR	IECR	<a href="#">Section 11.8.1.1.62</a>
105Ch	IECRH	IECRH	<a href="#">Section 11.8.1.1.63</a>
1060h	IESR	IESR	<a href="#">Section 11.8.1.1.64</a>
1064h	IESRH	IESRH	<a href="#">Section 11.8.1.1.65</a>
1068h	IPR	IPR	<a href="#">Section 11.8.1.1.66</a>
106Ch	IPRH	IPRH	<a href="#">Section 11.8.1.1.67</a>
1070h	ICR	ICR	<a href="#">Section 11.8.1.1.68</a>
1074h	ICRH	ICRH	<a href="#">Section 11.8.1.1.69</a>
1078h	IEVAL	IEVAL	<a href="#">Section 11.8.1.1.70</a>
1080h	QER	QER	<a href="#">Section 11.8.1.1.71</a>
1084h	QEER	QEER	<a href="#">Section 11.8.1.1.72</a>
1088h	QEECR	QEECR	<a href="#">Section 11.8.1.1.73</a>
108Ch	QEESR	QEESR	<a href="#">Section 11.8.1.1.74</a>
1090h	QSER	QSER	<a href="#">Section 11.8.1.1.75</a>
1094h	QSECR	QSECR	<a href="#">Section 11.8.1.1.76</a>
2000h	ER_RN	ER_RN	<a href="#">Section 11.8.1.1.77</a>
2004h	ERH_RN	ERH_RN	<a href="#">Section 11.8.1.1.78</a>
2008h	ECR_RN	ECR_RN	<a href="#">Section 11.8.1.1.79</a>
200Ch	ECRH_RN	ECRH_RN	<a href="#">Section 11.8.1.1.80</a>
2010h	ESR_RN	ESR_RN	<a href="#">Section 11.8.1.1.81</a>
2014h	ESRH_RN	ESRH_RN	<a href="#">Section 11.8.1.1.82</a>
2018h	CER_RN	CER_RN	<a href="#">Section 11.8.1.1.83</a>
201Ch	CERH_RN	CERH_RN	<a href="#">Section 11.8.1.1.84</a>
2020h	EER_RN	EER_RN	<a href="#">Section 11.8.1.1.85</a>
2024h	EERH_RN	EERH_RN	<a href="#">Section 11.8.1.1.86</a>

**Table 11-32. TPCC Registers (continued)**

Offset	Acronym	Register Name	Section
2028h	EECR_RN	EECR_RN	<a href="#">Section 11.8.1.1.87</a>
202Ch	EECRH_RN	EECRH_RN	<a href="#">Section 11.8.1.1.88</a>
2030h	EESR_RN	EESR_RN	<a href="#">Section 11.8.1.1.89</a>
2034h	EESRH_RN	EESRH_RN	<a href="#">Section 11.8.1.1.90</a>
2038h	SER_RN	SER_RN	<a href="#">Section 11.8.1.1.91</a>
203Ch	SERH_RN	SERH_RN	<a href="#">Section 11.8.1.1.92</a>
2040h	SECR_RN	SECR_RN	<a href="#">Section 11.8.1.1.93</a>
2044h	SECRH_RN	SECRH_RN	<a href="#">Section 11.8.1.1.94</a>
2050h	IER_RN	IER_RN	<a href="#">Section 11.8.1.1.95</a>
2054h	IERH_RN	IERH_RN	<a href="#">Section 11.8.1.1.96</a>
2058h	IECR_RN	IECR_RN	<a href="#">Section 11.8.1.1.97</a>
205Ch	IECRH_RN	IECRH_RN	<a href="#">Section 11.8.1.1.98</a>
2060h	IESR_RN	IESR_RN	<a href="#">Section 11.8.1.1.99</a>
2064h	IESRH_RN	IESRH_RN	<a href="#">Section 11.8.1.1.100</a>
2068h	IPR_RN	IPR_RN	<a href="#">Section 11.8.1.1.101</a>
206Ch	IPRH_RN	IPRH_RN	<a href="#">Section 11.8.1.1.102</a>
2070h	ICR_RN	ICR_RN	<a href="#">Section 11.8.1.1.103</a>
2074h	ICRH_RN	ICRH_RN	<a href="#">Section 11.8.1.1.104</a>
2078h	IEVAL_RN	IEVAL_RN	<a href="#">Section 11.8.1.1.105</a>
2080h	QER_RN	QER_RN	<a href="#">Section 11.8.1.1.106</a>
2084h	QEER_RN	QEER_RN	<a href="#">Section 11.8.1.1.107</a>
2088h	QEECR_RN	QEECR_RN	<a href="#">Section 11.8.1.1.108</a>
208Ch	QEESR_RN	QEESR_RN	<a href="#">Section 11.8.1.1.109</a>
2090h	QSER_RN	QSER_RN	<a href="#">Section 11.8.1.1.110</a>
2094h	QSECR_RN	QSECR_RN	<a href="#">Section 11.8.1.1.111</a>
4000h	OPT	OPT	<a href="#">Section 11.8.1.1.112</a>
4004h	SRC	SRC	<a href="#">Section 11.8.1.1.113</a>
4008h	ABCNT	ABCNT	<a href="#">Section 11.8.1.1.114</a>
400Ch	DST	DST	<a href="#">Section 11.8.1.1.115</a>
4010h	BIDX	BIDX	<a href="#">Section 11.8.1.1.116</a>
4014h	LNK	LNK	<a href="#">Section 11.8.1.1.117</a>
4018h	CIDX	CIDX	<a href="#">Section 11.8.1.1.118</a>
401Ch	CCNT	CCNT	<a href="#">Section 11.8.1.1.119</a>

### 11.8.1.1.1 PID Register (Offset = 0h) [reset = 4001AB00h]

PID is shown in [Figure 11-28](#) and described in [Table 11-33](#).

Return to the [Table 11-32](#).

Peripheral ID Register

**Figure 11-28. PID Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
SCHEME		RES1		FUNC											
R-1h		R-0h		R-1h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTL					MAJOR			CUSTOM		MINOR					
R-15h					R-3h			R-0h		R-0h					

**Table 11-33. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Scheme: Used to distinguish between old ID scheme and current. Spare bit to encode future schemes EDMA uses 'new scheme' indicated with value of 0x1.
29-28	RES1	R	0h	RESERVE FIELD
27-16	FUNC	R	1h	Function indicates a software compatible module family.
15-11	RTL	R	15h	RTL Version
10-8	MAJOR	R	3h	Major Revision
7-6	CUSTOM	R	0h	Custom revision field: Not used on this version of EDMA.
5-0	MINOR	R	0h	Minor Revision

### 11.8.1.1.2 CCCFG Register (Offset = 4h) [reset = 00213445h]

CCCFG is shown in [Figure 11-29](#) and described in [Table 11-34](#).

Return to the [Table 11-32](#).

CC Configuration Register

**Figure 11-29. CCCFG Register**

31	30	29	28	27	26	25	24
RES2						MPEXIST	CHMAPEXIST
R-0h						R-0h	R-0h
23	22	21	20	19	18	17	16
RES3		NUMREGN		RES4	NUMTC		
R-0h		R-2h		R-0h	R-1h		
15	14	13	12	11	10	9	8
RES5	NUMPAENTRY			RES6	NUMINTCH		
R-0h	R-3h			R-0h	R-4h		
7	6	5	4	3	2	1	0
RES7	NUMQDMACH			RES8	NUMDMACH		
R-0h	R-4h			R-0h	R-5h		

**Table 11-34. CCCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RES2	R	0h	RESERVE FIELD
25	MPEXIST	R	0h	Memory Protection Existence MPEXIST = 0 : No memory protection. MPEXIST = 1 : Memory Protection logic included.
24	CHMAPEXIST	R	0h	Channel Mapping Existence CHMAPEXIST = 0 : No Channel mapping. CHMAPEXIST = 1 : Channel mapping logic included.
23-22	RES3	R	0h	RESERVE FIELD
21-20	NUMREGN	R	2h	Number of MP and Shadow regions
19	RES4	R	0h	RESERVE FIELD
18-16	NUMTC	R	1h	Number of Queues/Number of TCs
15	RES5	R	0h	RESERVE FIELD
14-12	NUMPAENTRY	R	3h	Number of PaRAM entries
11	RES6	R	0h	RESERVE FIELD
10-8	NUMINTCH	R	4h	Number of Interrupt Channels
7	RES7	R	0h	RESERVE FIELD
6-4	NUMQDMACH	R	4h	Number of QDMA Channels
3	RES8	R	0h	RESERVE FIELD
2-0	NUMDMACH	R	5h	Number of DMA Channels

### 11.8.1.1.3 QCHMAPN Register (Offset = 200h) [reset = 0h]

QCHMAPN is shown in [Figure 11-30](#) and described in [Table 11-35](#).

Return to the [Table 11-32](#).

QDMA Channel N Mapping Register

**Figure 11-30. QCHMAPN Register**

31	30	29	28	27	26	25	24
RES10							
R-0h							
23	22	21	20	19	18	17	16
RES10							
R-0h							
15	14	13	12	11	10	9	8
RES10		PAENTRY					
R-0h		R/W-0h					
7	6	5	4	3	2	1	0
PAENTRY			TRWORD			RESERVED	
R/W-0h			R/W-0h			R-	

**Table 11-35. QCHMAPN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RES10	R	0h	RESERVE FIELD
13-5	PAENTRY	R/W	0h	PaRAM Entry number for QDMA Channel N.
4-2	TRWORD	R/W	0h	TRWORD points to the specific trigger word of the PaRAM Entry defined by PAENTRY. A write to the trigger word results in a QDMA Event being recognized.
1-0	RESERVED	R	0h	

#### 11.8.1.1.4 DMAQNUMN Register (Offset = 240h) [reset = 0h]

DMAQNUMN is shown in [Figure 11-31](#) and described in [Table 11-36](#).

Return to the [Table 11-32](#).

DMA Queue Number Register n Contains the Event queue number to be used for the corresponding DMA Channel.

**Figure 11-31. DMAQNUMN Register**

31	30	29	28	27	26	25	24
RES11	E7			RES12	E6		
R-0h			R/W-0h			R/W-0h	
23	22	21	20	19	18	17	16
RES13	E5			RES14	E4		
R-0h			R/W-0h			R/W-0h	
15	14	13	12	11	10	9	8
RES15	E3			RES16	E2		
R-0h			R/W-0h			R/W-0h	
7	6	5	4	3	2	1	0
RES17	E1			RES18	E0		
R-0h			R/W-0h			R/W-0h	

**Table 11-36. DMAQNUMN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RES11	R	0h	RESERVE FIELD
30-28	E7	R/W	0h	DMA Queue Number for event #7
27	RES12	R	0h	RESERVE FIELD
26-24	E6	R/W	0h	DMA Queue Number for event #6
23	RES13	R	0h	RESERVE FIELD
22-20	E5	R/W	0h	DMA Queue Number for event #5
19	RES14	R	0h	RESERVE FIELD
18-16	E4	R/W	0h	DMA Queue Number for event #4
15	RES15	R	0h	RESERVE FIELD
14-12	E3	R/W	0h	DMA Queue Number for event #3
11	RES16	R	0h	RESERVE FIELD
10-8	E2	R/W	0h	DMA Queue Number for event #2
7	RES17	R	0h	RESERVE FIELD
6-4	E1	R/W	0h	DMA Queue Number for event #1
3	RES18	R	0h	RESERVE FIELD
2-0	E0	R/W	0h	DMA Queue Number for event #0



### 11.8.1.1.5 QDMAQNUM Register (Offset = 260h) [reset = 0h]

QDMAQNUM is shown in [Figure 11-32](#) and described in [Table 11-37](#).

Return to the [Table 11-32](#).

QDMA Queue Number Register Contains the Event queue number to be used for the corresponding QDMA Channel.

**Figure 11-32. QDMAQNUM Register**

31	30	29	28	27	26	25	24
RES19	E7			RES20	E6		
R-0h	R/W-0h			R-0h	R/W-0h		
23	22	21	20	19	18	17	16
RES21	E5			RES22	E4		
R-0h	R/W-0h			R-0h	R/W-0h		
15	14	13	12	11	10	9	8
RES23	E3			RES24	E2		
R-0h	R/W-0h			R-0h	R/W-0h		
7	6	5	4	3	2	1	0
RES25	E1			RES26	E0		
R-0h	R/W-0h			R-0h	R/W-0h		

**Table 11-37. QDMAQNUM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	RES19	R	0h	RESERVE FIELD
30-28	E7	R/W	0h	QDMA Queue Number for event #7
27	RES20	R	0h	RESERVE FIELD
26-24	E6	R/W	0h	QDMA Queue Number for event #6
23	RES21	R	0h	RESERVE FIELD
22-20	E5	R/W	0h	QDMA Queue Number for event #5
19	RES22	R	0h	RESERVE FIELD
18-16	E4	R/W	0h	QDMA Queue Number for event #4
15	RES23	R	0h	RESERVE FIELD
14-12	E3	R/W	0h	QDMA Queue Number for event #3
11	RES24	R	0h	RESERVE FIELD
10-8	E2	R/W	0h	QDMA Queue Number for event #2
7	RES25	R	0h	RESERVE FIELD
6-4	E1	R/W	0h	QDMA Queue Number for event #1
3	RES26	R	0h	RESERVE FIELD
2-0	E0	R/W	0h	QDMA Queue Number for event #0

### 11.8.1.1.6 QUETCMAP Register (Offset = 280h) [reset = 10h]

QUETCMAP is shown in [Figure 11-33](#) and described in [Table 11-38](#).

Return to the [Table 11-32](#).

Queue to TC Mapping

**Figure 11-33. QUETCMAP Register**

31	30	29	28	27	26	25	24
RES27							
R-0h							
23	22	21	20	19	18	17	16
RES27							
R-0h							
15	14	13	12	11	10	9	8
RES27							
R-0h							
7	6	5	4	3	2	1	0
RES27	TCNUMQ1			RES28	TCNUMQ0		
R-0h	R/W-1h			R-0h	R/W-0h		

**Table 11-38. QUETCMAP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RES27	R	0h	RESERVE FIELD
6-4	TCNUMQ1	R/W	1h	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.
3	RES28	R	0h	RESERVE FIELD
2-0	TCNUMQ0	R/W	0h	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.

### 11.8.1.1.7 QUEPRI Register (Offset = 284h) [reset = 0h]

QUEPRI is shown in [Figure 11-34](#) and described in [Table 11-39](#).

Return to the [Table 11-32](#).

Queue Priority

**Figure 11-34. QUEPRI Register**

31	30	29	28	27	26	25	24
RES29							
R-0h							
23	22	21	20	19	18	17	16
RES29							
R-0h							
15	14	13	12	11	10	9	8
RES29							
R-0h							
7	6	5	4	3	2	1	0
RES29	PRIQ1			RES30	PRIQ0		
R-0h	R/W-0h			R-0h	R/W-0h		

**Table 11-39. QUEPRI Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RES29	R	0h	RESERVE FIELD
6-4	PRIQ1	R/W	0h	Priority Level for Queue 1 Dictates the priority level used for the OPTIONS field programming for Qn TRs. Sets the priority used for TC read and write commands.
3	RES30	R	0h	RESERVE FIELD
2-0	PRIQ0	R/W	0h	Priority Level for Queue 0 Dictates the priority level used for the OPTIONS field programming for Qn TRs. Sets the priority used for TC read and write commands.

### 11.8.1.1.8 EMR Register (Offset = 300h) [reset = 0h]

EMR is shown in [Figure 11-35](#) and described in [Table 11-40](#).

Return to the [Table 11-32](#).

Event Missed Register: The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER) Set Events (ESR) and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including QEMR/CCERR) were previously clear) then an error will be signaled with TPCC error interrupt.

**Figure 11-35. EMR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-40. EMR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event Missed #31
30	E30	R	0h	Event Missed #30
29	E29	R	0h	Event Missed #29
28	E28	R	0h	Event Missed #28
27	E27	R	0h	Event Missed #27
26	E26	R	0h	Event Missed #26
25	E25	R	0h	Event Missed #25
24	E24	R	0h	Event Missed #24
23	E23	R	0h	Event Missed #23
22	E22	R	0h	Event Missed #22
21	E21	R	0h	Event Missed #21
20	E20	R	0h	Event Missed #20
19	E19	R	0h	Event Missed #19
18	E18	R	0h	Event Missed #18
17	E17	R	0h	Event Missed #17
16	E16	R	0h	Event Missed #16
15	E15	R	0h	Event Missed #15
14	E14	R	0h	Event Missed #14
13	E13	R	0h	Event Missed #13
12	E12	R	0h	Event Missed #12
11	E11	R	0h	Event Missed #11
10	E10	R	0h	Event Missed #10
9	E9	R	0h	Event Missed #9
8	E8	R	0h	Event Missed #8
7	E7	R	0h	Event Missed #7

**Table 11-40. EMR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
6	E6	R	0h	Event Missed #6
5	E5	R	0h	Event Missed #5
4	E4	R	0h	Event Missed #4
3	E3	R	0h	Event Missed #3
2	E2	R	0h	Event Missed #2
1	E1	R	0h	Event Missed #1
0	E0	R	0h	Event Missed #0

### 11.8.1.1.9 EMRH Register (Offset = 304h) [reset = 0h]

EMRH is shown in [Figure 11-36](#) and described in [Table 11-41](#).

Return to the [Table 11-32](#).

Event Missed Register (High Part): The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER) Set Events (ESR) and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including QEMR/CCERR) were previously clear) then an error will be signaled with TPCC error interrupt.

**Figure 11-36. EMRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-41. EMRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event Missed #63
30	E62	R	0h	Event Missed #62
29	E61	R	0h	Event Missed #61
28	E60	R	0h	Event Missed #60
27	E59	R	0h	Event Missed #59
26	E58	R	0h	Event Missed #58
25	E57	R	0h	Event Missed #57
24	E56	R	0h	Event Missed #56
23	E55	R	0h	Event Missed #55
22	E54	R	0h	Event Missed #54
21	E53	R	0h	Event Missed #53
20	E52	R	0h	Event Missed #52
19	E51	R	0h	Event Missed #51
18	E50	R	0h	Event Missed #50
17	E49	R	0h	Event Missed #49
16	E48	R	0h	Event Missed #48
15	E47	R	0h	Event Missed #47
14	E46	R	0h	Event Missed #46
13	E45	R	0h	Event Missed #45
12	E44	R	0h	Event Missed #44
11	E43	R	0h	Event Missed #43
10	E42	R	0h	Event Missed #42
9	E41	R	0h	Event Missed #41
8	E40	R	0h	Event Missed #40
7	E39	R	0h	Event Missed #39

**Table 11-41. EMRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
6	E38	R	0h	Event Missed #38
5	E37	R	0h	Event Missed #37
4	E36	R	0h	Event Missed #36
3	E35	R	0h	Event Missed #35
2	E34	R	0h	Event Missed #34
1	E33	R	0h	Event Missed #33
0	E32	R	0h	Event Missed #32

### 11.8.1.1.10 EMCR Register (Offset = 308h) [reset = 0h]

EMCR is shown in [Figure 11-37](#) and described in [Table 11-42](#).

Return to the [Table 11-32](#).

Event Missed Clear Register: CPU write of '1' to the EMCR.En bit causes the EMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.

**Figure 11-37. EMCR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-42. EMCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event Missed Clear #31
30	E30	W	0h	Event Missed Clear #30
29	E29	W	0h	Event Missed Clear #29
28	E28	W	0h	Event Missed Clear #28
27	E27	W	0h	Event Missed Clear #27
26	E26	W	0h	Event Missed Clear #26
25	E25	W	0h	Event Missed Clear #25
24	E24	W	0h	Event Missed Clear #24
23	E23	W	0h	Event Missed Clear #23
22	E22	W	0h	Event Missed Clear #22
21	E21	W	0h	Event Missed Clear #21
20	E20	W	0h	Event Missed Clear #20
19	E19	W	0h	Event Missed Clear #19
18	E18	W	0h	Event Missed Clear #18
17	E17	W	0h	Event Missed Clear #17
16	E16	W	0h	Event Missed Clear #16
15	E15	W	0h	Event Missed Clear #15
14	E14	W	0h	Event Missed Clear #14
13	E13	W	0h	Event Missed Clear #13
12	E12	W	0h	Event Missed Clear #12
11	E11	W	0h	Event Missed Clear #11
10	E10	W	0h	Event Missed Clear #10
9	E9	W	0h	Event Missed Clear #9
8	E8	W	0h	Event Missed Clear #8
7	E7	W	0h	Event Missed Clear #7
6	E6	W	0h	Event Missed Clear #6



**Table 11-42. EMCR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event Missed Clear #5
4	E4	W	0h	Event Missed Clear #4
3	E3	W	0h	Event Missed Clear #3
2	E2	W	0h	Event Missed Clear #2
1	E1	W	0h	Event Missed Clear #1
0	E0	W	0h	Event Missed Clear #0

### 11.8.1.1.11 EMCRH Register (Offset = 30Ch) [reset = 0h]

EMCRH is shown in [Figure 11-38](#) and described in [Table 11-43](#).

Return to the [Table 11-32](#).

Event Missed Clear Register (High Part): CPU write of '1' to the EMCR.En bit causes the EMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.

**Figure 11-38. EMCRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-43. EMCRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event Missed Clear #63
30	E62	W	0h	Event Missed Clear #62
29	E61	W	0h	Event Missed Clear #61
28	E60	W	0h	Event Missed Clear #60
27	E59	W	0h	Event Missed Clear #59
26	E58	W	0h	Event Missed Clear #58
25	E57	W	0h	Event Missed Clear #57
24	E56	W	0h	Event Missed Clear #56
23	E55	W	0h	Event Missed Clear #55
22	E54	W	0h	Event Missed Clear #54
21	E53	W	0h	Event Missed Clear #53
20	E52	W	0h	Event Missed Clear #52
19	E51	W	0h	Event Missed Clear #51
18	E50	W	0h	Event Missed Clear #50
17	E49	W	0h	Event Missed Clear #49
16	E48	W	0h	Event Missed Clear #48
15	E47	W	0h	Event Missed Clear #47
14	E46	W	0h	Event Missed Clear #46
13	E45	W	0h	Event Missed Clear #45
12	E44	W	0h	Event Missed Clear #44
11	E43	W	0h	Event Missed Clear #43
10	E42	W	0h	Event Missed Clear #42
9	E41	W	0h	Event Missed Clear #41
8	E40	W	0h	Event Missed Clear #40
7	E39	W	0h	Event Missed Clear #39
6	E38	W	0h	Event Missed Clear #38

**Table 11-43. EMCRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event Missed Clear #37
4	E36	W	0h	Event Missed Clear #36
3	E35	W	0h	Event Missed Clear #35
2	E34	W	0h	Event Missed Clear #34
1	E33	W	0h	Event Missed Clear #33
0	E32	W	0h	Event Missed Clear #32

### 11.8.1.1.12 QEMR Register (Offset = 310h) [reset = 0h]

QEMR is shown in [Figure 11-39](#) and described in [Table 11-44](#).

Return to the [Table 11-32](#).

**QDMA Event Missed Register:** The QDMA Event Missed register is set if 2 QDMA events are detected without the first event being cleared or if a Null TR is serviced.. If any bit in the QEMR register is set (and all errors (including EMR/CCERR) were previously clear) then an error will be signaled with TPCC error interrupt.

**Figure 11-39. QEMR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES31															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES31								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-44. QEMR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES31	R	0h	RESERVE FIELD
7	E7	R	0h	Event Missed #7
6	E6	R	0h	Event Missed #6
5	E5	R	0h	Event Missed #5
4	E4	R	0h	Event Missed #4
3	E3	R	0h	Event Missed #3
2	E2	R	0h	Event Missed #2
1	E1	R	0h	Event Missed #1
0	E0	R	0h	Event Missed #0

### 11.8.1.1.13 QEMCR Register (Offset = 314h) [reset = 0h]

QEMCR is shown in [Figure 11-40](#) and described in [Table 11-45](#).

Return to the [Table 11-32](#).

QDMA Event Missed Clear Register: CPU write of '1' to the QEMCR.En bit causes the QEMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.

**Figure 11-40. QEMCR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES32															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES32								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-45. QEMCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES32	R	0h	RESERVE FIELD
7	E7	W	0h	Event Missed Clear #7
6	E6	W	0h	Event Missed Clear #6
5	E5	W	0h	Event Missed Clear #5
4	E4	W	0h	Event Missed Clear #4
3	E3	W	0h	Event Missed Clear #3
2	E2	W	0h	Event Missed Clear #2
1	E1	W	0h	Event Missed Clear #1
0	E0	W	0h	Event Missed Clear #0

### 11.8.1.1.14 CCERR Register (Offset = 318h) [reset = 0h]

CCERR is shown in [Figure 11-41](#) and described in [Table 11-46](#).

Return to the [Table 11-32](#).

CC Error Register

**Figure 11-41. CCERR Register**

31	30	29	28	27	26	25	24
RES33							
R-0h							
23	22	21	20	19	18	17	16
RES33						TCERR	
R-0h						R-0h	
15	14	13	12	11	10	9	8
RES34							
R-0h							
7	6	5	4	3	2	1	0
QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-46. CCERR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RES33	R	0h	RESERVE FIELD
16	TCERR	R	0h	Transfer Completion Code Error: TCCERR = 0 : Total number of allowed TCCs outstanding has not been reached. TCCERR = 1 : Total number of allowed TCCs has been reached. TCCERR can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors were previously clear) then an error will be signaled with TPCC error interrupt.
15-8	RES34	R	0h	RESERVE FIELD
7	QTHRXCD7	R	0h	Queue Threshold Error for Q7: QTHRXCD7 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD7 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD7 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
6	QTHRXCD6	R	0h	Queue Threshold Error for Q6: QTHRXCD6 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD6 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD6 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.

**Table 11-46. CCERR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	QTHRXCD5	R	0h	Queue Threshold Error for Q5: QTHRXCD5 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD5 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD5 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
4	QTHRXCD4	R	0h	Queue Threshold Error for Q4: QTHRXCD4 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD4 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD4 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
3	QTHRXCD3	R	0h	Queue Threshold Error for Q3: QTHRXCD3 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD3 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD3 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
2	QTHRXCD2	R	0h	Queue Threshold Error for Q2: QTHRXCD2 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD2 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD2 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
1	QTHRXCD1	R	0h	Queue Threshold Error for Q1: QTHRXCD1 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD1 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD1 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.
0	QTHRXCD0	R	0h	Queue Threshold Error for Q0: QTHRXCD0 = 0 : Watermark/ threshold has not been exceeded. QTHRXCD0 = 1 : Watermark/ threshold has been exceeded. CCERR.QTHRXCD0 can be cleared by writing a '1' to corresponding bit in CCERRCLR register. If any bit in the CCERR register is set (and all errors (including EMR/QEMR) were previously clear) then an error will be signaled with the TPCC error interrupt.

### 11.8.1.1.15 CCERRCLR Register (Offset = 31Ch) [reset = 0h]

CCERRCLR is shown in [Figure 11-42](#) and described in [Table 11-47](#).

Return to the [Table 11-32](#).

CC Error Clear Register

**Figure 11-42. CCERRCLR Register**

31	30	29	28	27	26	25	24
RES35							
R-0h							
23	22	21	20	19	18	17	16
RES35						TCERR	
R-0h						W-0h	
15	14	13	12	11	10	9	8
RES36							
R-0h							
7	6	5	4	3	2	1	0
QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-47. CCERRCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	RES35	R	0h	RESERVE FIELD
16	TCERR	W	0h	Clear Error for CCERR.TCERR: Write of '1' clears the value of CCERR bit N. Writes of '0' have no affect.
15-8	RES36	R	0h	RESERVE FIELD
7	QTHRXCD7	W	0h	Clear error for CCERR.QTHRXCD7: Write of '1' clears the values of QSTAT7.WM QSTAT7.THRXCD CCERR.QTHRXCD7 Writes of '0' have no affect.
6	QTHRXCD6	W	0h	Clear error for CCERR.QTHRXCD6: Write of '1' clears the values of QSTAT6.WM QSTAT6.THRXCD CCERR.QTHRXCD6 Writes of '0' have no affect.
5	QTHRXCD5	W	0h	Clear error for CCERR.QTHRXCD5: Write of '1' clears the values of QSTAT5.WM QSTAT5.THRXCD CCERR.QTHRXCD5 Writes of '0' have no affect.
4	QTHRXCD4	W	0h	Clear error for CCERR.QTHRXCD4: Write of '1' clears the values of QSTAT4.WM QSTAT4.THRXCD CCERR.QTHRXCD4 Writes of '0' have no affect.
3	QTHRXCD3	W	0h	Clear error for CCERR.QTHRXCD3: Write of '1' clears the values of QSTAT3.WM QSTAT3.THRXCD CCERR.QTHRXCD3 Writes of '0' have no affect.
2	QTHRXCD2	W	0h	Clear error for CCERR.QTHRXCD2: Write of '1' clears the values of QSTAT2.WM QSTAT2.THRXCD CCERR.QTHRXCD2 Writes of '0' have no affect.
1	QTHRXCD1	W	0h	Clear error for CCERR.QTHRXCD1: Write of '1' clears the values of QSTAT1.WM QSTAT1.THRXCD CCERR.QTHRXCD1 Writes of '0' have no affect.



**Table 11-47. CCERRCLR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	QTHRXCD0	W	0h	Clear error for CCERR.QTHRXCD0: Write of '1' clears the values of QSTAT0.WM QSTAT0.THRXCD CCERR.QTHRXCD0 Writes of '0' have no affect.

### 11.8.1.1.16 EEVAL Register (Offset = 320h) [reset = 0h]

EEVAL is shown in [Figure 11-43](#) and described in [Table 11-48](#).

Return to the [Table 11-32](#).

Error Eval Register

**Figure 11-43. EEVAL Register**

31	30	29	28	27	26	25	24
RES37							
R-0h							
23	22	21	20	19	18	17	16
RES37							
R-0h							
15	14	13	12	11	10	9	8
RES37							
R-0h							
7	6	5	4	3	2	1	0
RES37						SET	EVAL
R-0h						W-0h	W-0h

**Table 11-48. EEVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RES37	R	0h	RESERVE FIELD
1	SET	W	0h	Error Interrupt Set: CPU write of '1' to the SET bit causes the TPCC error interrupt to be pulsed regardless of state of EMR/EMRH QEMR or CCERR. CPU write of '0' has no effect.
0	EVAL	W	0h	Error Interrupt Evaluate: CPU write of '1' to the EVAL bit causes the TPCC error interrupt to be pulsed if any errors have not been cleared in the EMR/EMRH QEMR or CCERR registers. CPU write of '0' has no effect.

### 11.8.1.1.17 DRAEM Register (Offset = 340h) [reset = 0h]

DRAEM is shown in [Figure 11-44](#) and described in [Table 11-49](#).

Return to the [Table 11-32](#).

DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.

**Figure 11-44. DRAEM Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**Table 11-49. DRAEM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R/W	0h	DMA Region Access enable for Region M bit #31
30	E30	R/W	0h	DMA Region Access enable for Region M bit #30
29	E29	R/W	0h	DMA Region Access enable for Region M bit #29
28	E28	R/W	0h	DMA Region Access enable for Region M bit #28
27	E27	R/W	0h	DMA Region Access enable for Region M bit #27
26	E26	R/W	0h	DMA Region Access enable for Region M bit #26
25	E25	R/W	0h	DMA Region Access enable for Region M bit #25
24	E24	R/W	0h	DMA Region Access enable for Region M bit #24
23	E23	R/W	0h	DMA Region Access enable for Region M bit #23
22	E22	R/W	0h	DMA Region Access enable for Region M bit #22
21	E21	R/W	0h	DMA Region Access enable for Region M bit #21
20	E20	R/W	0h	DMA Region Access enable for Region M bit #20
19	E19	R/W	0h	DMA Region Access enable for Region M bit #19
18	E18	R/W	0h	DMA Region Access enable for Region M bit #18
17	E17	R/W	0h	DMA Region Access enable for Region M bit #17
16	E16	R/W	0h	DMA Region Access enable for Region M bit #16
15	E15	R/W	0h	DMA Region Access enable for Region M bit #15
14	E14	R/W	0h	DMA Region Access enable for Region M bit #14
13	E13	R/W	0h	DMA Region Access enable for Region M bit #13
12	E12	R/W	0h	DMA Region Access enable for Region M bit #12
11	E11	R/W	0h	DMA Region Access enable for Region M bit #11
10	E10	R/W	0h	DMA Region Access enable for Region M bit #10
9	E9	R/W	0h	DMA Region Access enable for Region M bit #9

**Table 11-49. DRAEM Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R/W	0h	DMA Region Access enable for Region M bit #8
7	E7	R/W	0h	DMA Region Access enable for Region M bit #7
6	E6	R/W	0h	DMA Region Access enable for Region M bit #6
5	E5	R/W	0h	DMA Region Access enable for Region M bit #5
4	E4	R/W	0h	DMA Region Access enable for Region M bit #4
3	E3	R/W	0h	DMA Region Access enable for Region M bit #3
2	E2	R/W	0h	DMA Region Access enable for Region M bit #2
1	E1	R/W	0h	DMA Region Access enable for Region M bit #1
0	E0	R/W	0h	DMA Region Access enable for Region M bit #0

### 11.8.1.1.18 DRAEHM Register (Offset = 344h) [reset = 0h]

DRAEHM is shown in [Figure 11-45](#) and described in [Table 11-50](#).

Return to the [Table 11-32](#).

DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt. En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.

**Figure 11-45. DRAEHM Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**Table 11-50. DRAEHM Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R/W	0h	DMA Region Access enable for Region M bit #63
30	E62	R/W	0h	DMA Region Access enable for Region M bit #62
29	E61	R/W	0h	DMA Region Access enable for Region M bit #61
28	E60	R/W	0h	DMA Region Access enable for Region M bit #60
27	E59	R/W	0h	DMA Region Access enable for Region M bit #59
26	E58	R/W	0h	DMA Region Access enable for Region M bit #58
25	E57	R/W	0h	DMA Region Access enable for Region M bit #57
24	E56	R/W	0h	DMA Region Access enable for Region M bit #56
23	E55	R/W	0h	DMA Region Access enable for Region M bit #55
22	E54	R/W	0h	DMA Region Access enable for Region M bit #54
21	E53	R/W	0h	DMA Region Access enable for Region M bit #53
20	E52	R/W	0h	DMA Region Access enable for Region M bit #52
19	E51	R/W	0h	DMA Region Access enable for Region M bit #51
18	E50	R/W	0h	DMA Region Access enable for Region M bit #50
17	E49	R/W	0h	DMA Region Access enable for Region M bit #49
16	E48	R/W	0h	DMA Region Access enable for Region M bit #48
15	E47	R/W	0h	DMA Region Access enable for Region M bit #47
14	E46	R/W	0h	DMA Region Access enable for Region M bit #46
13	E45	R/W	0h	DMA Region Access enable for Region M bit #45
12	E44	R/W	0h	DMA Region Access enable for Region M bit #44

**Table 11-50. DRAEHM Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
11	E43	R/W	0h	DMA Region Access enable for Region M bit #43
10	E42	R/W	0h	DMA Region Access enable for Region M bit #42
9	E41	R/W	0h	DMA Region Access enable for Region M bit #41
8	E40	R/W	0h	DMA Region Access enable for Region M bit #40
7	E39	R/W	0h	DMA Region Access enable for Region M bit #39
6	E38	R/W	0h	DMA Region Access enable for Region M bit #38
5	E37	R/W	0h	DMA Region Access enable for Region M bit #37
4	E36	R/W	0h	DMA Region Access enable for Region M bit #36
3	E35	R/W	0h	DMA Region Access enable for Region M bit #35
2	E34	R/W	0h	DMA Region Access enable for Region M bit #34
1	E33	R/W	0h	DMA Region Access enable for Region M bit #33
0	E32	R/W	0h	DMA Region Access enable for Region M bit #32

### 11.8.1.1.19 QRAEN Register (Offset = 380h) [reset = 0h]

QRAEN is shown in [Figure 11-46](#) and described in [Table 11-51](#).

Return to the [Table 11-32](#).

QDMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any QDMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any QDMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region n interrupt.

**Figure 11-46. QRAEN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES38															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES38								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**Table 11-51. QRAEN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES38	R	0h	RESERVE FIELD
7	E7	R/W	0h	QDMA Region Access enable for Region M bit #7
6	E6	R/W	0h	QDMA Region Access enable for Region M bit #6
5	E5	R/W	0h	QDMA Region Access enable for Region M bit #5
4	E4	R/W	0h	QDMA Region Access enable for Region M bit #4
3	E3	R/W	0h	QDMA Region Access enable for Region M bit #3
2	E2	R/W	0h	QDMA Region Access enable for Region M bit #2
1	E1	R/W	0h	QDMA Region Access enable for Region M bit #1
0	E0	R/W	0h	QDMA Region Access enable for Region M bit #0

### 11.8.1.1.20 QNE0 Register (Offset = 400h) [reset = 0h]

QNE0 is shown in [Figure 11-47](#) and described in [Table 11-52](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 0

**Figure 11-47. QNE0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES39															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES39								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-52. QNE0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES39	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).



### 11.8.1.1.21 QNE1 Register (Offset = 404h) [reset = 0h]

QNE1 is shown in [Figure 11-48](#) and described in [Table 11-53](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 1

**Figure 11-48. QNE1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES40															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES40								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-53. QNE1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES40	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.22 QNE2 Register (Offset = 408h) [reset = 0h]

QNE2 is shown in [Figure 11-49](#) and described in [Table 11-54](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 2

**Figure 11-49. QNE2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES41															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES41								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-54. QNE2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES41	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.23 QNE3 Register (Offset = 40Ch) [reset = 0h]

QNE3 is shown in [Figure 11-50](#) and described in [Table 11-55](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 3

**Figure 11-50. QNE3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES42															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES42								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-55. QNE3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES42	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.24 QNE4 Register (Offset = 410h) [reset = 0h]

QNE4 is shown in [Figure 11-51](#) and described in [Table 11-56](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 4

**Figure 11-51. QNE4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES43															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES43								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-56. QNE4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES43	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.25 QNE5 Register (Offset = 414h) [reset = 0h]

QNE5 is shown in [Figure 11-52](#) and described in [Table 11-57](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 5

**Figure 11-52. QNE5 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES44															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES44								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-57. QNE5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES44	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.26 QNE6 Register (Offset = 418h) [reset = 0h]

QNE6 is shown in [Figure 11-53](#) and described in [Table 11-58](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 6

**Figure 11-53. QNE6 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES45															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES45								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-58. QNE6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES45	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.27 QNE7 Register (Offset = 41Ch) [reset = 0h]

QNE7 is shown in [Figure 11-54](#) and described in [Table 11-59](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 7

**Figure 11-54. QNE7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES46															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES46								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-59. QNE7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES46	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.28 QNE8 Register (Offset = 420h) [reset = 0h]

QNE8 is shown in [Figure 11-55](#) and described in [Table 11-60](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 8

**Figure 11-55. QNE8 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES47															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES47								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-60. QNE8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES47	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).



### 11.8.1.1.29 QNE9 Register (Offset = 424h) [reset = 0h]

QNE9 is shown in [Figure 11-56](#) and described in [Table 11-61](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 9

**Figure 11-56. QNE9 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES48															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES48								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-61. QNE9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES48	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.30 QNE10 Register (Offset = 428h) [reset = 0h]

QNE10 is shown in [Figure 11-57](#) and described in [Table 11-62](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 0

**Figure 11-57. QNE10 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES49															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES49								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-62. QNE10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES49	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.31 QNE11 Register (Offset = 42Ch) [reset = 0h]

QNE11 is shown in [Figure 11-58](#) and described in [Table 11-63](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 11

**Figure 11-58. QNE11 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES50															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES50								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-63. QNE11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES50	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.32 QNE12 Register (Offset = 430h) [reset = 0h]

QNE12 is shown in [Figure 11-59](#) and described in [Table 11-64](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 12

**Figure 11-59. QNE12 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES51															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES51								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-64. QNE12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES51	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.33 QNE13 Register (Offset = 434h) [reset = 0h]

QNE13 is shown in [Figure 11-60](#) and described in [Table 11-65](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 13

**Figure 11-60. QNE13 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES52															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES52								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-65. QNE13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES52	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.34 QNE14 Register (Offset = 438h) [reset = 0h]

QNE14 is shown in [Figure 11-61](#) and described in [Table 11-66](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 14

**Figure 11-61. QNE14 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES53															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES53								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-66. QNE14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES53	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.35 QNE15 Register (Offset = 43Ch) [reset = 0h]

QNE15 is shown in [Figure 11-62](#) and described in [Table 11-67](#).

Return to the [Table 11-32](#).

Event Queue Entry Diagram for Queue n - Entry 15

**Figure 11-62. QNE15 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES54															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES54								ETYPE		ENUM					
R-0h								R-0h		R-0h					

**Table 11-67. QNE15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES54	R	0h	RESERVE FIELD
7-6	ETYPE	R	0h	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.
5-0	ENUM	R	0h	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (ER/ESR/CER) ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (QER) ENUM will range between 0 and NUM_QDMACH (up to 7).

### 11.8.1.1.36 QSTATN Register (Offset = 600h) [reset = 0h]

QSTATN is shown in [Figure 11-63](#) and described in [Table 11-68](#).

Return to the [Table 11-32](#).

QSTATn Register Set

**Figure 11-63. QSTATN Register**

31	30	29	28	27	26	25	24
RES55							THRXCD
R-0h							R-0h
23	22	21	20	19	18	17	16
RES56				WM			
R-0h				R-0h			
15	14	13	12	11	10	9	8
RES57				NUMVAL			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RES58				STRTPTR			
R-0h				R-0h			

**Table 11-68. QSTATN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RES55	R	0h	RESERVE FIELD
24	THRXCD	R	0h	Threshold Exceeded: THRXCD = 0 : Threshold specified by QWMTHR(A B).Qn has not been exceeded. THRXCD = 1 : Threshold specified by QWMTHR(A B).Qn has been exceeded. QSTATn.THXXCD is cleared via CCERR.WMCLRN bit.
23-21	RES56	R	0h	RESERVE FIELD
20-16	WM	R	0h	Watermark for Maximum Queue Usage: Watermark tracks the most entries that have been in QueueN since reset or since the last time that the watermark (WM) was cleared. QSTATn.WM is cleared via CCERR.WMCLRN bit. Legal values = 0x0 (empty) to 0x10 (full)
15-13	RES57	R	0h	RESERVE FIELD
12-8	NUMVAL	R	0h	Number of Valid Entries in QueueN: Represents the total number of entries residing in the Queue Manager FIFO at a given instant. Always enabled. Legal values = 0x0 (empty) to 0x10 (full)
7-4	RES58	R	0h	RESERVE FIELD
3-0	STRTPTR	R	0h	Start Pointer: Represents the offset to the head entry of QueueN in units of entries. Always enabled. Legal values = 0x0 (0th entry) to 0xF (15th entry)



### 11.8.1.1.37 QWMTHRA Register (Offset = 620h) [reset = 1010h]

QWMTHRA is shown in [Figure 11-64](#) and described in [Table 11-69](#).

Return to the [Table 11-32](#).

Queue Threshold A for Q[3:0]: CCERR.QTHRXCdN and QSTATn.THRXCD error bit is set when the number of Events in QueueN at an instant in time (visible via QSTATn.NUMVAL) equals or exceeds the value specified by QWMTHRA.Qn. Legal values = 0x0 (ever used?) to 0x10 (ever full?) A value of 0x11 disables threshold errors.

**Figure 11-64. QWMTHRA Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES59															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES59				Q1				RES60				Q0			
R-0h				R/W-10h				R-0h				R/W-10h			

**Table 11-69. QWMTHRA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RES59	R	0h	RESERVE FIELD
12-8	Q1	R/W	10h	Queue Threshold for Q1 value
7-5	RES60	R	0h	RESERVE FIELD
4-0	Q0	R/W	10h	Queue Threshold for Q0 value

### 11.8.1.1.38 CCSTAT Register (Offset = 640h) [reset = 0h]

CCSTAT is shown in [Figure 11-65](#) and described in [Table 11-70](#).

Return to the [Table 11-32](#).

CC Status Register

**Figure 11-65. CCSTAT Register**

31	30	29	28	27	26	25	24
RES61							
R-0h							
23	22	21	20	19	18	17	16
QUEACTV7	QUEACTV6	QUEACTV5	QUEACTV4	QUEACTV3	QUEACTV2	QUEACTV1	QUEACTV0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RES62		COMPACTV					
R-0h		R-0h					
7	6	5	4	3	2	1	0
RES63			ACTV	RES64	TRACTV	QEV TACTV	EVTACTV
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-70. CCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RES61	R	0h	RESERVE FIELD
23	QUEACTV7	R	0h	Queue 7 Active QUEACTV7 = 0 : No Evts are queued in Q7. QUEACTV7 = 1 : At least one TR is queued in Q7.
22	QUEACTV6	R	0h	Queue 6 Active QUEACTV6 = 0 : No Evts are queued in Q6. QUEACTV6 = 1 : At least one TR is queued in Q6.
21	QUEACTV5	R	0h	Queue 5 Active QUEACTV5 = 0 : No Evts are queued in Q5. QUEACTV5 = 1 : At least one TR is queued in Q5.
20	QUEACTV4	R	0h	Queue 4 Active QUEACTV4 = 0 : No Evts are queued in Q4. QUEACTV4 = 1 : At least one TR is queued in Q4.
19	QUEACTV3	R	0h	Queue 3 Active QUEACTV3 = 0 : No Evts are queued in Q3. QUEACTV3 = 1 : At least one TR is queued in Q3.
18	QUEACTV2	R	0h	Queue 2 Active QUEACTV2 = 0 : No Evts are queued in Q2. QUEACTV2 = 1 : At least one TR is queued in Q2.
17	QUEACTV1	R	0h	Queue 1 Active QUEACTV1 = 0 : No Evts are queued in Q1. QUEACTV1 = 1 : At least one TR is queued in Q1.
16	QUEACTV0	R	0h	Queue 0 Active QUEACTV0 = 0 : No Evts are queued in Q0. QUEACTV0 = 1 : At least one TR is queued in Q0.
15-14	RES62	R	0h	RESERVE FIELD

**Table 11-70. CCSTAT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
13-8	COMPACTV	R	0h	Completion Request Active: Counter that tracks the total number of completion requests submitted to the TC. The counter increments when a TR is submitted with TCINTEN or TCCHEN set to '1'. The counter decrements for every valid completion code received from any of the external TCs. The CC will not service new TRs if COMPACTV count is already at the limit. COMPACTV = 0 : No completion requests outstanding. COMPACTV = 1: Total of '1' completion request outstanding. ... COMPACTV = 63 : Total of 63 completion requests are outstanding. No additional TRs will be submitted until count is less than 63.
7-5	RES63	R	0h	RESERVE FIELD
4	ACTV	R	0h	Channel Controller Active: Channel Controller Active is a logical-OR of each of the ACTV signals. The ACTV bit must remain high through the life of a TR. ACTV = 0 : Channel is idle. ACTV = 1 : Channel is busy.
3	RES64	R	0h	RESERVE FIELD
2	TRACTV	R	0h	Transfer Request Active: TRACTV = 0 : Transfer Request processing/submission logic is inactive. TRACTV = 1 : Transfer Request processing/submission logic is active.
1	QEVACTV	R	0h	QDMA Event Active: QEVACTV = 0 : No enabled QDMA Events are active within the CC. QEVACTV = 1 : At least one enabled DMA Event (ER & EER ESR CER) is active within the CC.
0	EVTACTV	R	0h	DMA Event Active: EVTACTV = 0 : No enabled DMA Events are active within the CC. EVTACTV = 1 : At least one enabled DMA Event (ER & EER ESR CER) is active within the CC.

### 11.8.1.1.39 AETCTL Register (Offset = 700h) [reset = 0h]

AETCTL is shown in [Figure 11-66](#) and described in [Table 11-71](#).

Return to the [Table 11-32](#).

Advanced Event Trigger Control

**Figure 11-66. AETCTL Register**

31	30	29	28	27	26	25	24
EN	RES65						
R/W-0h	R-0h						
23	22	21	20	19	18	17	16
RES65							
R-0h							
15	14	13	12	11	10	9	8
RES65		ENDINT					
R-0h		R/W-0h					
7	6	5	4	3	2	1	0
RES66	TYPE	STRTEVT					
R-0h	R/W-0h	R/W-0h					

**Table 11-71. AETCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	EN	R/W	0h	AET Enable: EN = 0 : AET event generation is disabled. EN = 1 : AET event generation is enabled.
30-14	RES65	R	0h	RESERVE FIELD
13-8	ENDINT	R/W	0h	AET End Interrupt: Dictates the completion interrupt number that will force the tpcc_aet signal to be deasserted (low)
7	RES66	R	0h	RESERVE FIELD
6	TYPE	R/W	0h	AET Event Type: TYPE = 0 : Event specified by STARTEVT applies to DMA Events (set by ER ESR or CER) TYPE = 1 : Event specified by STARTEVT applies to QDMA Events
5-0	STRTEVT	R/W	0h	AET Start Event: Dictates the Event Number that will force the tpcc_aet signal to be asserted (high)

#### 11.8.1.1.40 AETSTAT Register (Offset = 704h) [reset = 0h]

AETSTAT is shown in [Figure 11-67](#) and described in [Table 11-72](#).

Return to the [Table 11-32](#).

Advanced Event Trigger Stat

**Figure 11-67. AETSTAT Register**

31	30	29	28	27	26	25	24
RES67							
R-0h							
23	22	21	20	19	18	17	16
RES67							
R-0h							
15	14	13	12	11	10	9	8
RES67							
R-0h							
7	6	5	4	3	2	1	0
RES67							STAT
R-0h							R-0h

**Table 11-72. AETSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RES67	R	0h	RESERVE FIELD
0	STAT	R	0h	AET Status: AETSTAT = 0 : tpcc_aet is currently low. AETSTAT = 1 : tpcc_aet is currently high.

#### 11.8.1.1.41 AETCMD Register (Offset = 708h) [reset = 0h]

AETCMD is shown in [Figure 11-68](#) and described in [Table 11-73](#).

Return to the [Table 11-32](#).

AET Command

**Figure 11-68. AETCMD Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES68															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES68															CLR
R-0h															W-0h

**Table 11-73. AETCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RES68	R	0h	RESERVE FIELD
0	CLR	W	0h	AET Clear command: CPU write of '1' to the CLR bit causes the tpcc_aet output signal and AETSTAT.STAT register to be cleared. CPU write of '0' has no effect..

### 11.8.1.1.42 ER Register (Offset = 1000h) [reset = 0h]

ER is shown in [Figure 11-69](#) and described in [Table 11-74](#).

Return to the [Table 11-32](#).

Event Register: If ER.En bit is set and the EER.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ER.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EER.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EER register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECR pseudo-register.

**Figure 11-69. ER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-74. ER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9

**Table 11-74. ER Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0



### 11.8.1.1.43 ERH Register (Offset = 1004h) [reset = 0h]

ERH is shown in [Figure 11-70](#) and described in [Table 11-75](#).

Return to the [Table 11-32](#).

Event Register (High Part): If ERH.En bit is set and the EERH.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ERH.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EERH.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EERH register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECRH pseudo-register.

**Figure 11-70. ERH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-75. ERH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41

**Table 11-75. ERH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

#### 11.8.1.1.44 ECR Register (Offset = 1008h) [reset = 0h]

ECR is shown in [Figure 11-71](#) and described in [Table 11-76](#).

Return to the [Table 11-32](#).

Event Clear Register: CPU write of '1' to the ECR.En bit causes the ER.En bit to be cleared. CPU write of '0' has no effect.

**Figure 11-71. ECR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-76. ECR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-76. ECR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.45 ECRH Register (Offset = 100Ch) [reset = 0h]

ECRH is shown in [Figure 11-72](#) and described in [Table 11-77](#).

Return to the [Table 11-32](#).

Event Clear Register (High Part): CPU write of '1' to the ECRH.En bit causes the ERH.En bit to be cleared. CPU write of '0' has no effect.

**Figure 11-72. ECRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-77. ECRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-77. ECRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.46 ESR Register (Offset = 1010h) [reset = 0h]

ESR is shown in [Figure 11-73](#) and described in [Table 11-78](#).

Return to the [Table 11-32](#).

Event Set Register: CPU write of '1' to the ESR.En bit causes the ER.En bit to be set. CPU write of '0' has no effect.

**Figure 11-73. ESR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-78. ESR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-78. ESR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0



### 11.8.1.1.47 ESRH Register (Offset = 1014h) [reset = 0h]

ESRH is shown in [Figure 11-74](#) and described in [Table 11-79](#).

Return to the [Table 11-32](#).

Event Set Register (High Part) CPU write of '1' to the ESRH.En bit causes the ERH.En bit to be set. CPU write of '0' has no effect.

**Figure 11-74. ESRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-79. ESRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-79. ESRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.48 CER Register (Offset = 1018h) [reset = 0h]

CER is shown in [Figure 11-75](#) and described in [Table 11-80](#).

Return to the [Table 11-32](#).

Chained Event Register: If CER.En bit is set (regardless of state of EER.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CER.En bit is cleared when the corresponding event is prioritized and serviced. If the CER.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CER.En cannot be set or cleared via software.

**Figure 11-75. CER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-80. CER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9

**Table 11-80. CER Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.49 CERH Register (Offset = 101Ch) [reset = 0h]

CERH is shown in [Figure 11-76](#) and described in [Table 11-81](#).

Return to the [Table 11-32](#).

Chained Event Register (High Part): If CERH.En bit is set (regardless of state of EERH.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the CERH.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CERH.En cannot be set or cleared via software.

**Figure 11-76. CERH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-81. CERH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41

**Table 11-81. CERH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.50 EER Register (Offset = 1020h) [reset = 0h]

EER is shown in [Figure 11-77](#) and described in [Table 11-82](#).

Return to the [Table 11-32](#).

Event Enable Register: Enables DMA transfers for ER.En pending events. ER.En is set based on externally asserted events (via tpcc\_eventN\_pi). This register has no effect on Chained Event Register (CER) or Event Set Register (ESR). Note that if a bit is set in ER.En while EER.En is disabled no action is taken. If EER.En is enabled at a later point (and ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EER.En is not directly writeable. Events can be enabled via writes to EESR and can be disabled via writes to EECR register. EER.En = 0: ER.En is not enabled to trigger DMA transfers. EER.En = 1: ER.En is enabled to trigger DMA transfers.

**Figure 11-77. EER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-82. EER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9

**Table 11-82. EER Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0



### 11.8.1.1.51 EERH Register (Offset = 1024h) [reset = 0h]

EERH is shown in [Figure 11-78](#) and described in [Table 11-83](#).

Return to the [Table 11-32](#).

Event Enable Register (High Part): Enables DMA transfers for ERH.En pending events. ERH.En is set based on externally asserted events (via tpcc\_eventN\_pi). This register has no effect on Chained Event Register (CERH) or Event Set Register (ESRH). Note that if a bit is set in ERH.En while EERH.En is disabled no action is taken. If EERH.En is enabled at a later point (and ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EERH.En is not directly writeable. Events can be enabled via writes to EESRH and can be disabled via writes to EECRH register. EERH.En = 0: ER.En is not enabled to trigger DMA transfers. EERH.En = 1: ER.En is enabled to trigger DMA transfers.

**Figure 11-78. EERH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-83. EERH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41

**Table 11-83. EERH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.52 EECR Register (Offset = 1028h) [reset = 0h]

EECR is shown in [Figure 11-79](#) and described in [Table 11-84](#).

Return to the [Table 11-32](#).

Event Enable Clear Register: CPU write of '1' to the EECR.En bit causes the EER.En bit to be cleared. CPU write of '0' has no effect..

**Figure 11-79. EECR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-84. EECR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-84. EECR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.53 EECRH Register (Offset = 102Ch) [reset = 0h]

EECRH is shown in [Figure 11-80](#) and described in [Table 11-85](#).

Return to the [Table 11-32](#).

Event Enable Clear Register (High Part): CPU write of '1' to the EECRH.En bit causes the EERH.En bit to be cleared. CPU write of '0' has no effect..

**Figure 11-80. EECRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-85. EECRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-85. EECRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.54 EESR Register (Offset = 1030h) [reset = 0h]

EESR is shown in [Figure 11-81](#) and described in [Table 11-86](#).

Return to the [Table 11-32](#).

Event Enable Set Register: CPU write of '1' to the EESR.En bit causes the EER.En bit to be set. CPU write of '0' has no effect..

**Figure 11-81. EESR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-86. EESR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-86. EESR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0



### 11.8.1.1.55 EESRH Register (Offset = 1034h) [reset = 0h]

EESRH is shown in [Figure 11-82](#) and described in [Table 11-87](#).

Return to the [Table 11-32](#).

Event Enable Set Register (High Part): CPU write of '1' to the EESRH.En bit causes the EERH.En bit to be set. CPU write of '0' has no effect..

**Figure 11-82. EESRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-87. EESRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-87. EESRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.56 SER Register (Offset = 1038h) [reset = 0h]

SER is shown in [Figure 11-83](#) and described in [Table 11-88](#).

Return to the [Table 11-32](#).

Secondary Event Register: The secondary event register is used along with the Event Register (ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

**Figure 11-83. SER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-88. SER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6

**Table 11-88. SER Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.57 SERH Register (Offset = 103Ch) [reset = 0h]

SERH is shown in [Figure 11-84](#) and described in [Table 11-89](#).

Return to the [Table 11-32](#).

Secondary Event Register (High Part): The secondary event register is used along with the Event Register (ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

**Figure 11-84. SERH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-89. SERH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38

**Table 11-89. SERH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.58 SECR Register (Offset = 1040h) [reset = 0h]

SECR is shown in [Figure 11-85](#) and described in [Table 11-90](#).

Return to the [Table 11-32](#).

Secondary Event Clear Register: The secondary event clear register is used to clear the status of the SER registers. CPU write of '1' to the SECR.En bit clears the SER register. CPU write of '0' has no effect.

**Figure 11-85. SECR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-90. SECR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-90. SECR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0



### 11.8.1.1.59 SECRH Register (Offset = 1044h) [reset = 0h]

SECRH is shown in [Figure 11-86](#) and described in [Table 11-91](#).

Return to the [Table 11-32](#).

Secondary Event Clear Register (High Part): The secondary event clear register is used to clear the status of the SERH registers. CPU write of '1' to the SECRH.En bit clears the SERH register. CPU write of '0' has no effect.

**Figure 11-86. SECRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-91. SECRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-91. SECRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.60 IER Register (Offset = 1050h) [reset = 0h]

IER is shown in [Figure 11-87](#) and described in [Table 11-92](#).

Return to the [Table 11-32](#).

Int Enable Register: IER.In is not directly writeable. Interrupts can be enabled via writes to IESR and can be disabled via writes to IECR register. IER.In = 0: IPR.In is NOT enabled for interrupts. IER.In = 1: IPR.In IS enabled for interrupts.

**Figure 11-87. IER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-92. IER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6

**Table 11-92. IER Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	R	0h	Interrupt associated with TCC #5
4	I4	R	0h	Interrupt associated with TCC #4
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

### 11.8.1.1.61 IERH Register (Offset = 1054h) [reset = 0h]

IERH is shown in [Figure 11-88](#) and described in [Table 11-93](#).

Return to the [Table 11-32](#).

Int Enable Register (High Part): IERH.In is not directly writeable. Interrupts can be enabled via writes to IESRH and can be disabled via writes to IECRH register. IERH.In = 0: IPRH.In is NOT enabled for interrupts. IERH.In = 1: IPRH.In IS enabled for interrupts.

**Figure 11-88. IERH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-93. IERH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38

**Table 11-93. IERH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	R	0h	Interrupt associated with TCC #37
4	I36	R	0h	Interrupt associated with TCC #36
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

### 11.8.1.1.62 IECR Register (Offset = 1058h) [reset = 0h]

IECR is shown in [Figure 11-89](#) and described in [Table 11-94](#).

Return to the [Table 11-32](#).

Int Enable Clear Register: CPU write of '1' to the IECR.In bit causes the IER.In bit to be cleared. CPU write of '0' has no effect..

**Figure 11-89. IECR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-94. IECR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6

**Table 11-94. IECR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0



### 11.8.1.1.63 IECRH Register (Offset = 105Ch) [reset = 0h]

IECRH is shown in [Figure 11-90](#) and described in [Table 11-95](#).

Return to the [Table 11-32](#).

Int Enable Clear Register (High Part): CPU write of '1' to the IECRH.In bit causes the IERH.In bit to be cleared. CPU write of '0' has no effect..

**Figure 11-90. IECRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-95. IECRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38

**Table 11-95. IECRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

### 11.8.1.1.64 IESR Register (Offset = 1060h) [reset = 0h]

IESR is shown in [Figure 11-91](#) and described in [Table 11-96](#).

Return to the [Table 11-32](#).

Int Enable Set Register: CPU write of '1' to the IESR.In bit causes the IESR.In bit to be set. CPU write of '0' has no effect..

**Figure 11-91. IESR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-96. IESR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6

**Table 11-96. IESR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

### 11.8.1.1.65 IESRH Register (Offset = 1064h) [reset = 0h]

IESRH is shown in [Figure 11-92](#) and described in [Table 11-97](#).

Return to the [Table 11-32](#).

Int Enable Set Register (High Part): CPU write of '1' to the IESRH.In bit causes the IESRH.In bit to be set. CPU write of '0' has no effect..

**Figure 11-92. IESRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-97. IESRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38

**Table 11-97. IESRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

### 11.8.1.1.66 IPR Register (Offset = 1068h) [reset = 0h]

IPR is shown in [Figure 11-93](#) and described in [Table 11-98](#).

Return to the [Table 11-32](#).

Interrupt Pending Register: IPR.In bit is set when a interrupt completion code with TCC of N is detected. IPR.In bit is cleared via software by writing a '1' to ICR.In bit.

**Figure 11-93. IPR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-98. IPR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6

**Table 11-98. IPR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	R	0h	Interrupt associated with TCC #5
4	I4	R	0h	Interrupt associated with TCC #4
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0



### 11.8.1.1.67 IPRH Register (Offset = 106Ch) [reset = 0h]

IPRH is shown in [Figure 11-94](#) and described in [Table 11-99](#).

Return to the [Table 11-32](#).

Interrupt Pending Register (High Part): IPRH.In bit is set when a interrupt completion code with TCC of N is detected. IPRH.In bit is cleared via software by writing a '1' to ICRH.In bit.

**Figure 11-94. IPRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-99. IPRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38

**Table 11-99. IPRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	R	0h	Interrupt associated with TCC #37
4	I36	R	0h	Interrupt associated with TCC #36
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

### 11.8.1.1.68 ICR Register (Offset = 1070h) [reset = 0h]

ICR is shown in [Figure 11-95](#) and described in [Table 11-100](#).

Return to the [Table 11-32](#).

Interrupt Clear Register: CPU write of '1' to the ICR.In bit causes the IPR.In bit to be cleared. CPU write of '0' has no effect. All IPR.In bits must be cleared before additional interrupts will be asserted by CC.

**Figure 11-95. ICR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-100. ICR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6

**Table 11-100. ICR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

### 11.8.1.1.69 ICRH Register (Offset = 1074h) [reset = 0h]

ICRH is shown in [Figure 11-96](#) and described in [Table 11-101](#).

Return to the [Table 11-32](#).

Interrupt Clear Register (High Part): CPU write of '1' to the ICRH.In bit causes the IPRH.In bit to be cleared. CPU write of '0' has no effect. All IPRH.In bits must be cleared before additional interrupts will be asserted by CC.

**Figure 11-96. ICRH Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-101. ICRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38

**Table 11-101. ICRH Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

### 11.8.1.1.70 IEVAL Register (Offset = 1078h) [reset = 0h]

IEVAL is shown in [Figure 11-97](#) and described in [Table 11-102](#).

Return to the [Table 11-32](#).

Interrupt Eval Register

**Figure 11-97. IEVAL Register**

31	30	29	28	27	26	25	24
RES69							
R-0h							
23	22	21	20	19	18	17	16
RES69							
R-0h							
15	14	13	12	11	10	9	8
RES69							
R-0h							
7	6	5	4	3	2	1	0
RES69						SET	EVAL
R-0h						W-0h	W-0h

**Table 11-102. IEVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RES69	R	0h	RESERVE FIELD
1	SET	W	0h	Interrupt Set: CPU write of '1' to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (IPRn). CPU write of '0' has no effect.
0	EVAL	W	0h	Interrupt Evaluate: CPU write of '1' to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (IPRn). CPU write of '0' has no effect..

### 11.8.1.1.71 QER Register (Offset = 1080h) [reset = 0h]

QER is shown in [Figure 11-98](#) and described in [Table 11-103](#).

Return to the [Table 11-32](#).

QDMA Event Register: If QER.En bit is set then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. QER.En bit is cleared when the corresponding event is prioritized and serviced. QER.En is also cleared when user writes a '1' to the QSECR.En bit. If the QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and QEER register is set then the corresponding bit in the QDMA Event Missed Register is set.

**Figure 11-98. QER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES70															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES70								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-103. QER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES70	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0



### 11.8.1.1.72 QEER Register (Offset = 1084h) [reset = 0h]

QEER is shown in [Figure 11-99](#) and described in [Table 11-104](#).

Return to the [Table 11-32](#).

QDMA Event Enable Register: Enabled/disabled QDMA address comparator for QDMA Channel N. QEER.En is not directly writeable. QDMA channels can be enabled via writes to QEESR and can be disabled via writes to QEECR register. QEER.En = 1 The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in QER.En. QEER.En = 0 The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in QER.En.

**Figure 11-99. QEER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES71															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES71								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-104. QEER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES71	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.73 QEECR Register (Offset = 1088h) [reset = 0h]

QEECR is shown in [Figure 11-100](#) and described in [Table 11-105](#).

Return to the [Table 11-32](#).

QDMA Event Enable Clear Register: CPU write of '1' to the QEECR.En bit causes the QEER.En bit to be cleared. CPU write of '0' has no effect..

**Figure 11-100. QEECR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES72															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES72								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-105. QEECR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES72	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

#### 11.8.1.1.74 QEESR Register (Offset = 108Ch) [reset = 0h]

QEESR is shown in [Figure 11-101](#) and described in [Table 11-106](#).

Return to the [Table 11-32](#).

QDMA Event Enable Set Register: CPU write of '1' to the QEESR.En bit causes the QEESR.En bit to be set. CPU write of '0' has no effect..

**Figure 11-101. QEESR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES73															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES73								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-106. QEESR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES73	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.75 QSER Register (Offset = 1090h) [reset = 0h]

QSER is shown in [Figure 11-102](#) and described in [Table 11-107](#).

Return to the [Table 11-32](#).

QDMA Secondary Event Register: The QDMA secondary event register is used along with the QDMA Event Register (QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

**Figure 11-102. QSER Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES74															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES74								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-107. QSER Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES74	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.76 QSECR Register (Offset = 1094h) [reset = 0h]

QSECR is shown in [Figure 11-103](#) and described in [Table 11-108](#).

Return to the [Table 11-32](#).

QDMA Secondary Event Clear Register: The secondary event clear register is used to clear the status of the QSER and QER register (note that this is slightly different than the SER operation which does not clear the ER.En register). CPU write of '1' to the QSECR.En bit clears the QSER.En and QER.En register fields. CPU write of '0' has no effect..

**Figure 11-103. QSECR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES75															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES75								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-108. QSECR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES75	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.77 ER\_RN Register (Offset = 2000h) [reset = 0h]

ER\_RN is shown in [Figure 11-104](#) and described in [Table 11-109](#).

Return to the [Table 11-32](#).

Event Register: If ER.En bit is set and the EER.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ER.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EER.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EER register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECR pseudo-register.

**Figure 11-104. ER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-109. ER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9

**Table 11-109. ER\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.78 ERH\_RN Register (Offset = 2004h) [reset = 0h]

ERH\_RN is shown in [Figure 11-105](#) and described in [Table 11-110](#).

Return to the [Table 11-32](#).

Event Register (High Part): If ERH.En bit is set and the EERH.En bit is also set then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. ERH.En bit is set when the input event #n transitions from inactive (low) to active (high) regardless of the state of EERH.En bit. ER.En bit is cleared when the corresponding event is prioritized and serviced. If the ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EERH register is set then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the ECRH pseudo-register.

**Figure 11-105. ERH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-110. ERH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41



**Table 11-110. ERH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.79 ECR\_RN Register (Offset = 2008h) [reset = 0h]

ECR\_RN is shown in [Figure 11-106](#) and described in [Table 11-111](#).

Return to the [Table 11-32](#).

Event Clear Register: CPU write of '1' to the ECR.En bit causes the ER.En bit to be cleared. CPU write of '0' has no effect.

**Figure 11-106. ECR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-111. ECR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-111. ECR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.80 ECRH\_RN Register (Offset = 200Ch) [reset = 0h]

ECRH\_RN is shown in [Figure 11-107](#) and described in [Table 11-112](#).

Return to the [Table 11-32](#).

Event Clear Register (High Part): CPU write of '1' to the ECRH.En bit causes the ERH.En bit to be cleared. CPU write of '0' has no effect.

**Figure 11-107. ECRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-112. ECRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-112. ECRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.81 ESR\_RN Register (Offset = 2010h) [reset = 0h]

ESR\_RN is shown in [Figure 11-108](#) and described in [Table 11-113](#).

Return to the [Table 11-32](#).

Event Set Register: CPU write of '1' to the ESR.En bit causes the ER.En bit to be set. CPU write of '0' has no effect.

**Figure 11-108. ESR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-113. ESR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-113. ESR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.82 ESRH\_RN Register (Offset = 2014h) [reset = 0h]

ESRH\_RN is shown in [Figure 11-109](#) and described in [Table 11-114](#).

Return to the [Table 11-32](#).

Event Set Register (High Part) CPU write of '1' to the ESRH.En bit causes the ERH.En bit to be set. CPU write of '0' has no effect.

**Figure 11-109. ESRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-114. ESRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38



**Table 11-114. ESRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.83 CER\_RN Register (Offset = 2018h) [reset = 0h]

CER\_RN is shown in [Figure 11-110](#) and described in [Table 11-115](#).

Return to the [Table 11-32](#).

Chained Event Register: If CER.En bit is set (regardless of state of EER.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CER.En bit is cleared when the corresponding event is prioritized and serviced. If the CER.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CER.En cannot be set or cleared via software.

**Figure 11-110. CER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-115. CER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9

**Table 11-115. CER\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.84 CERH\_RN Register (Offset = 201Ch) [reset = 0h]

CERH\_RN is shown in [Figure 11-111](#) and described in [Table 11-116](#).

Return to the [Table 11-32](#).

Chained Event Register (High Part): If CERH.En bit is set (regardless of state of EERH.En) then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface or is generated internally via Early Completion path. CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the CERH.En bit is already set and the corresponding chaining completion code is returned from the TC then the corresponding bit in the Event Missed Register is set. CERH.En cannot be set or cleared via software.

**Figure 11-111. CERH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-116. CERH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41

**Table 11-116. CERH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.85 EER\_RN Register (Offset = 2020h) [reset = 0h]

EER\_RN is shown in [Figure 11-112](#) and described in [Table 11-117](#).

Return to the [Table 11-32](#).

Event Enable Register: Enables DMA transfers for ER.En pending events. ER.En is set based on externally asserted events (via tpcc\_eventN\_pi). This register has no effect on Chained Event Register (CER) or Event Set Register (ESR). Note that if a bit is set in ER.En while EER.En is disabled no action is taken. If EER.En is enabled at a later point (and ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EER.En is not directly writeable. Events can be enabled via writes to EESR and can be disabled via writes to EECR register. EER.En = 0: ER.En is not enabled to trigger DMA transfers. EER.En = 1: ER.En is enabled to trigger DMA transfers.

**Figure 11-112. EER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-117. EER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9

**Table 11-117. EER\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.86 EERH\_RN Register (Offset = 2024h) [reset = 0h]

EERH\_RN is shown in [Figure 11-113](#) and described in [Table 11-118](#).

Return to the [Table 11-32](#).

Event Enable Register (High Part): Enables DMA transfers for ERH.En pending events. ERH.En is set based on externally asserted events (via tpcc\_eventN\_pi). This register has no effect on Chained Event Register (CERH) or Event Set Register (ESRH). Note that if a bit is set in ERH.En while EERH.En is disabled no action is taken. If EERH.En is enabled at a later point (and ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EERH.En is not directly writeable. Events can be enabled via writes to EESRH and can be disabled via writes to EECRH register. EERH.En = 0: ER.En is not enabled to trigger DMA transfers. EERH.En = 1: ER.En is enabled to trigger DMA transfers.

**Figure 11-113. EERH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-118. EERH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41



**Table 11-118. EERH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.87 EECR\_RN Register (Offset = 2028h) [reset = 0h]

EECR\_RN is shown in [Figure 11-114](#) and described in [Table 11-119](#).

Return to the [Table 11-32](#).

Event Enable Clear Register: CPU write of '1' to the EECR.En bit causes the EER.En bit to be cleared. CPU write of '0' has no effect..

**Figure 11-114. EECR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-119. EECR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-119. EECR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.88 EECRH\_RN Register (Offset = 202Ch) [reset = 0h]

EECRH\_RN is shown in [Figure 11-115](#) and described in [Table 11-120](#).

Return to the [Table 11-32](#).

Event Enable Clear Register (High Part): CPU write of '1' to the EECRH.En bit causes the EERH.En bit to be cleared. CPU write of '0' has no effect..

**Figure 11-115. EECRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-120. EECRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38

**Table 11-120. EECRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.89 EESR\_RN Register (Offset = 2030h) [reset = 0h]

EESR\_RN is shown in [Figure 11-116](#) and described in [Table 11-121](#).

Return to the [Table 11-32](#).

Event Enable Set Register: CPU write of '1' to the EESR.En bit causes the EER.En bit to be set. CPU write of '0' has no effect..

**Figure 11-116. EESR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-121. EESR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-121. EESR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.90 EESRH\_RN Register (Offset = 2034h) [reset = 0h]

EESRH\_RN is shown in [Figure 11-117](#) and described in [Table 11-122](#).

Return to the [Table 11-32](#).

Event Enable Set Register (High Part): CPU write of '1' to the EESRH.En bit causes the EERH.En bit to be set. CPU write of '0' has no effect..

**Figure 11-117. EESRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-122. EESRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38



**Table 11-122. EESRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.91 SER\_RN Register (Offset = 2038h) [reset = 0h]

SER\_RN is shown in [Figure 11-118](#) and described in [Table 11-123](#).

Return to the [Table 11-32](#).

Secondary Event Register: The secondary event register is used along with the Event Register (ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

**Figure 11-118. SER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-123. SER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	R	0h	Event #31
30	E30	R	0h	Event #30
29	E29	R	0h	Event #29
28	E28	R	0h	Event #28
27	E27	R	0h	Event #27
26	E26	R	0h	Event #26
25	E25	R	0h	Event #25
24	E24	R	0h	Event #24
23	E23	R	0h	Event #23
22	E22	R	0h	Event #22
21	E21	R	0h	Event #21
20	E20	R	0h	Event #20
19	E19	R	0h	Event #19
18	E18	R	0h	Event #18
17	E17	R	0h	Event #17
16	E16	R	0h	Event #16
15	E15	R	0h	Event #15
14	E14	R	0h	Event #14
13	E13	R	0h	Event #13
12	E12	R	0h	Event #12
11	E11	R	0h	Event #11
10	E10	R	0h	Event #10
9	E9	R	0h	Event #9
8	E8	R	0h	Event #8
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6

**Table 11-123. SER\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.92 SERH\_RN Register (Offset = 203Ch) [reset = 0h]

SERH\_RN is shown in [Figure 11-119](#) and described in [Table 11-124](#).

Return to the [Table 11-32](#).

Secondary Event Register (High Part): The secondary event register is used along with the Event Register (ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

**Figure 11-119. SERH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-124. SERH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	R	0h	Event #63
30	E62	R	0h	Event #62
29	E61	R	0h	Event #61
28	E60	R	0h	Event #60
27	E59	R	0h	Event #59
26	E58	R	0h	Event #58
25	E57	R	0h	Event #57
24	E56	R	0h	Event #56
23	E55	R	0h	Event #55
22	E54	R	0h	Event #54
21	E53	R	0h	Event #53
20	E52	R	0h	Event #52
19	E51	R	0h	Event #51
18	E50	R	0h	Event #50
17	E49	R	0h	Event #49
16	E48	R	0h	Event #48
15	E47	R	0h	Event #47
14	E46	R	0h	Event #46
13	E45	R	0h	Event #45
12	E44	R	0h	Event #44
11	E43	R	0h	Event #43
10	E42	R	0h	Event #42
9	E41	R	0h	Event #41
8	E40	R	0h	Event #40
7	E39	R	0h	Event #39
6	E38	R	0h	Event #38

**Table 11-124. SERH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	R	0h	Event #37
4	E36	R	0h	Event #36
3	E35	R	0h	Event #35
2	E34	R	0h	Event #34
1	E33	R	0h	Event #33
0	E32	R	0h	Event #32

### 11.8.1.1.93 SECR\_RN Register (Offset = 2040h) [reset = 0h]

SECR\_RN is shown in [Figure 11-120](#) and described in [Table 11-125](#).

Return to the [Table 11-32](#).

Secondary Event Clear Register: The secondary event clear register is used to clear the status of the SER registers. CPU write of '1' to the SECR.En bit clears the SER register. CPU write of '0' has no effect.

**Figure 11-120. SECR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-125. SECR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E31	W	0h	Event #31
30	E30	W	0h	Event #30
29	E29	W	0h	Event #29
28	E28	W	0h	Event #28
27	E27	W	0h	Event #27
26	E26	W	0h	Event #26
25	E25	W	0h	Event #25
24	E24	W	0h	Event #24
23	E23	W	0h	Event #23
22	E22	W	0h	Event #22
21	E21	W	0h	Event #21
20	E20	W	0h	Event #20
19	E19	W	0h	Event #19
18	E18	W	0h	Event #18
17	E17	W	0h	Event #17
16	E16	W	0h	Event #16
15	E15	W	0h	Event #15
14	E14	W	0h	Event #14
13	E13	W	0h	Event #13
12	E12	W	0h	Event #12
11	E11	W	0h	Event #11
10	E10	W	0h	Event #10
9	E9	W	0h	Event #9
8	E8	W	0h	Event #8
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6

**Table 11-125. SECR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.94 SECRH\_RN Register (Offset = 2044h) [reset = 0h]

SECRH\_RN is shown in [Figure 11-121](#) and described in [Table 11-126](#).

Return to the [Table 11-32](#).

Secondary Event Clear Register (High Part): The secondary event clear register is used to clear the status of the SERH registers. CPU write of '1' to the SECRH.En bit clears the SERH register. CPU write of '0' has no effect.

**Figure 11-121. SECRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-126. SECRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	E63	W	0h	Event #63
30	E62	W	0h	Event #62
29	E61	W	0h	Event #61
28	E60	W	0h	Event #60
27	E59	W	0h	Event #59
26	E58	W	0h	Event #58
25	E57	W	0h	Event #57
24	E56	W	0h	Event #56
23	E55	W	0h	Event #55
22	E54	W	0h	Event #54
21	E53	W	0h	Event #53
20	E52	W	0h	Event #52
19	E51	W	0h	Event #51
18	E50	W	0h	Event #50
17	E49	W	0h	Event #49
16	E48	W	0h	Event #48
15	E47	W	0h	Event #47
14	E46	W	0h	Event #46
13	E45	W	0h	Event #45
12	E44	W	0h	Event #44
11	E43	W	0h	Event #43
10	E42	W	0h	Event #42
9	E41	W	0h	Event #41
8	E40	W	0h	Event #40
7	E39	W	0h	Event #39
6	E38	W	0h	Event #38



**Table 11-126. SECRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	E37	W	0h	Event #37
4	E36	W	0h	Event #36
3	E35	W	0h	Event #35
2	E34	W	0h	Event #34
1	E33	W	0h	Event #33
0	E32	W	0h	Event #32

### 11.8.1.1.95 IER\_RN Register (Offset = 2050h) [reset = 0h]

IER\_RN is shown in [Figure 11-122](#) and described in [Table 11-127](#).

Return to the [Table 11-32](#).

Int Enable Register: IER.In is not directly writeable. Interrupts can be enabled via writes to IESR and can be disabled via writes to IECR register. IER.In = 0: IPR.In is NOT enabled for interrupts. IER.In = 1: IPR.In IS enabled for interrupts.

**Figure 11-122. IER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-127. IER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6

**Table 11-127. IER\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	R	0h	Interrupt associated with TCC #5
4	I4	R	0h	Interrupt associated with TCC #4
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

### 11.8.1.1.96 IERH\_RN Register (Offset = 2054h) [reset = 0h]

IERH\_RN is shown in [Figure 11-123](#) and described in [Table 11-128](#).

Return to the [Table 11-32](#).

Int Enable Register (High Part): IERH.In is not directly writeable. Interrupts can be enabled via writes to IESRH and can be disabled via writes to IECRH register. IERH.In = 0: IPRH.In is NOT enabled for interrupts. IERH.In = 1: IPRH.In IS enabled for interrupts.

**Figure 11-123. IERH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-128. IERH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38

**Table 11-128. IERH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	R	0h	Interrupt associated with TCC #37
4	I36	R	0h	Interrupt associated with TCC #36
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

### 11.8.1.1.97 IECR\_RN Register (Offset = 2058h) [reset = 0h]

IECR\_RN is shown in [Figure 11-124](#) and described in [Table 11-129](#).

Return to the [Table 11-32](#).

Int Enable Clear Register: CPU write of '1' to the IECR.In bit causes the IER.In bit to be cleared. CPU write of '0' has no effect..

**Figure 11-124. IECR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-129. IECR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6

**Table 11-129. IECR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

### 11.8.1.1.98 IECRH\_RN Register (Offset = 205Ch) [reset = 0h]

IECRH\_RN is shown in [Figure 11-125](#) and described in [Table 11-130](#).

Return to the [Table 11-32](#).

Int Enable Clear Register (High Part): CPU write of '1' to the IECRH.In bit causes the IERH.In bit to be cleared. CPU write of '0' has no effect..

**Figure 11-125. IECRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-130. IECRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38



**Table 11-130. IECRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

### 11.8.1.1.99 IESR\_RN Register (Offset = 2060h) [reset = 0h]

IESR\_RN is shown in [Figure 11-126](#) and described in [Table 11-131](#).

Return to the [Table 11-32](#).

Int Enable Set Register: CPU write of '1' to the IESR.In bit causes the IESR.In bit to be set. CPU write of '0' has no effect..

**Figure 11-126. IESR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-131. IESR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6

**Table 11-131. IESR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

### 11.8.1.1.100 IESRH\_RN Register (Offset = 2064h) [reset = 0h]

IESRH\_RN is shown in [Figure 11-127](#) and described in [Table 11-132](#).

Return to the [Table 11-32](#).

Int Enable Set Register (High Part): CPU write of '1' to the IESRH.In bit causes the IESRH.In bit to be set. CPU write of '0' has no effect..

**Figure 11-127. IESRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-132. IESRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38

**Table 11-132. IESRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

### 11.8.1.1.101 IPR\_RN Register (Offset = 2068h) [reset = 0h]

IPR\_RN is shown in [Figure 11-128](#) and described in [Table 11-133](#).

Return to the [Table 11-32](#).

Interrupt Pending Register: IPR.In bit is set when a interrupt completion code with TCC of N is detected. IPR.In bit is cleared via software by writing a '1' to ICR.In bit.

**Figure 11-128. IPR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-133. IPR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	R	0h	Interrupt associated with TCC #31
30	I30	R	0h	Interrupt associated with TCC #30
29	I29	R	0h	Interrupt associated with TCC #29
28	I28	R	0h	Interrupt associated with TCC #28
27	I27	R	0h	Interrupt associated with TCC #27
26	I26	R	0h	Interrupt associated with TCC #26
25	I25	R	0h	Interrupt associated with TCC #25
24	I24	R	0h	Interrupt associated with TCC #24
23	I23	R	0h	Interrupt associated with TCC #23
22	I22	R	0h	Interrupt associated with TCC #22
21	I21	R	0h	Interrupt associated with TCC #21
20	I20	R	0h	Interrupt associated with TCC #20
19	I19	R	0h	Interrupt associated with TCC #19
18	I18	R	0h	Interrupt associated with TCC #18
17	I17	R	0h	Interrupt associated with TCC #17
16	I16	R	0h	Interrupt associated with TCC #16
15	I15	R	0h	Interrupt associated with TCC #15
14	I14	R	0h	Interrupt associated with TCC #14
13	I13	R	0h	Interrupt associated with TCC #13
12	I12	R	0h	Interrupt associated with TCC #12
11	I11	R	0h	Interrupt associated with TCC #11
10	I10	R	0h	Interrupt associated with TCC #10
9	I9	R	0h	Interrupt associated with TCC #9
8	I8	R	0h	Interrupt associated with TCC #8
7	I7	R	0h	Interrupt associated with TCC #7
6	I6	R	0h	Interrupt associated with TCC #6

**Table 11-133. IPR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	R	0h	Interrupt associated with TCC #5
4	I4	R	0h	Interrupt associated with TCC #4
3	I3	R	0h	Interrupt associated with TCC #3
2	I2	R	0h	Interrupt associated with TCC #2
1	I1	R	0h	Interrupt associated with TCC #1
0	I0	R	0h	Interrupt associated with TCC #0

### 11.8.1.1.102 IPRH\_RN Register (Offset = 206Ch) [reset = 0h]

IPRH\_RN is shown in [Figure 11-129](#) and described in [Table 11-134](#).

Return to the [Table 11-32](#).

Interrupt Pending Register (High Part): IPRH.In bit is set when a interrupt completion code with TCC of N is detected. IPRH.In bit is cleared via software by writing a '1' to ICRH.In bit.

**Figure 11-129. IPRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-134. IPRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	R	0h	Interrupt associated with TCC #63
30	I62	R	0h	Interrupt associated with TCC #62
29	I61	R	0h	Interrupt associated with TCC #61
28	I60	R	0h	Interrupt associated with TCC #60
27	I59	R	0h	Interrupt associated with TCC #59
26	I58	R	0h	Interrupt associated with TCC #58
25	I57	R	0h	Interrupt associated with TCC #57
24	I56	R	0h	Interrupt associated with TCC #56
23	I55	R	0h	Interrupt associated with TCC #55
22	I54	R	0h	Interrupt associated with TCC #54
21	I53	R	0h	Interrupt associated with TCC #53
20	I52	R	0h	Interrupt associated with TCC #52
19	I51	R	0h	Interrupt associated with TCC #51
18	I50	R	0h	Interrupt associated with TCC #50
17	I49	R	0h	Interrupt associated with TCC #49
16	I48	R	0h	Interrupt associated with TCC #48
15	I47	R	0h	Interrupt associated with TCC #47
14	I46	R	0h	Interrupt associated with TCC #46
13	I45	R	0h	Interrupt associated with TCC #45
12	I44	R	0h	Interrupt associated with TCC #44
11	I43	R	0h	Interrupt associated with TCC #43
10	I42	R	0h	Interrupt associated with TCC #42
9	I41	R	0h	Interrupt associated with TCC #41
8	I40	R	0h	Interrupt associated with TCC #40
7	I39	R	0h	Interrupt associated with TCC #39
6	I38	R	0h	Interrupt associated with TCC #38



**Table 11-134. IPRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	R	0h	Interrupt associated with TCC #37
4	I36	R	0h	Interrupt associated with TCC #36
3	I35	R	0h	Interrupt associated with TCC #35
2	I34	R	0h	Interrupt associated with TCC #34
1	I33	R	0h	Interrupt associated with TCC #33
0	I32	R	0h	Interrupt associated with TCC #32

### 11.8.1.1.103 ICR\_RN Register (Offset = 2070h) [reset = 0h]

ICR\_RN is shown in [Figure 11-130](#) and described in [Table 11-135](#).

Return to the [Table 11-32](#).

Interrupt Clear Register: CPU write of '1' to the ICR.In bit causes the IPR.In bit to be cleared. CPU write of '0' has no effect. All IPR.In bits must be cleared before additional interrupts will be asserted by CC.

**Figure 11-130. ICR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-135. ICR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I31	W	0h	Interrupt associated with TCC #31
30	I30	W	0h	Interrupt associated with TCC #30
29	I29	W	0h	Interrupt associated with TCC #29
28	I28	W	0h	Interrupt associated with TCC #28
27	I27	W	0h	Interrupt associated with TCC #27
26	I26	W	0h	Interrupt associated with TCC #26
25	I25	W	0h	Interrupt associated with TCC #25
24	I24	W	0h	Interrupt associated with TCC #24
23	I23	W	0h	Interrupt associated with TCC #23
22	I22	W	0h	Interrupt associated with TCC #22
21	I21	W	0h	Interrupt associated with TCC #21
20	I20	W	0h	Interrupt associated with TCC #20
19	I19	W	0h	Interrupt associated with TCC #19
18	I18	W	0h	Interrupt associated with TCC #18
17	I17	W	0h	Interrupt associated with TCC #17
16	I16	W	0h	Interrupt associated with TCC #16
15	I15	W	0h	Interrupt associated with TCC #15
14	I14	W	0h	Interrupt associated with TCC #14
13	I13	W	0h	Interrupt associated with TCC #13
12	I12	W	0h	Interrupt associated with TCC #12
11	I11	W	0h	Interrupt associated with TCC #11
10	I10	W	0h	Interrupt associated with TCC #10
9	I9	W	0h	Interrupt associated with TCC #9
8	I8	W	0h	Interrupt associated with TCC #8
7	I7	W	0h	Interrupt associated with TCC #7
6	I6	W	0h	Interrupt associated with TCC #6

**Table 11-135. ICR\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I5	W	0h	Interrupt associated with TCC #5
4	I4	W	0h	Interrupt associated with TCC #4
3	I3	W	0h	Interrupt associated with TCC #3
2	I2	W	0h	Interrupt associated with TCC #2
1	I1	W	0h	Interrupt associated with TCC #1
0	I0	W	0h	Interrupt associated with TCC #0

### 11.8.1.1.104 ICRH\_RN Register (Offset = 2074h) [reset = 0h]

ICRH\_RN is shown in [Figure 11-131](#) and described in [Table 11-136](#).

Return to the [Table 11-32](#).

Interrupt Clear Register (High Part): CPU write of '1' to the ICRH.In bit causes the IPRH.In bit to be cleared. CPU write of '0' has no effect. All IPRH.In bits must be cleared before additional interrupts will be asserted by CC.

**Figure 11-131. ICRH\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32
W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-136. ICRH\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	I63	W	0h	Interrupt associated with TCC #63
30	I62	W	0h	Interrupt associated with TCC #62
29	I61	W	0h	Interrupt associated with TCC #61
28	I60	W	0h	Interrupt associated with TCC #60
27	I59	W	0h	Interrupt associated with TCC #59
26	I58	W	0h	Interrupt associated with TCC #58
25	I57	W	0h	Interrupt associated with TCC #57
24	I56	W	0h	Interrupt associated with TCC #56
23	I55	W	0h	Interrupt associated with TCC #55
22	I54	W	0h	Interrupt associated with TCC #54
21	I53	W	0h	Interrupt associated with TCC #53
20	I52	W	0h	Interrupt associated with TCC #52
19	I51	W	0h	Interrupt associated with TCC #51
18	I50	W	0h	Interrupt associated with TCC #50
17	I49	W	0h	Interrupt associated with TCC #49
16	I48	W	0h	Interrupt associated with TCC #48
15	I47	W	0h	Interrupt associated with TCC #47
14	I46	W	0h	Interrupt associated with TCC #46
13	I45	W	0h	Interrupt associated with TCC #45
12	I44	W	0h	Interrupt associated with TCC #44
11	I43	W	0h	Interrupt associated with TCC #43
10	I42	W	0h	Interrupt associated with TCC #42
9	I41	W	0h	Interrupt associated with TCC #41
8	I40	W	0h	Interrupt associated with TCC #40
7	I39	W	0h	Interrupt associated with TCC #39
6	I38	W	0h	Interrupt associated with TCC #38

**Table 11-136. ICRH\_RN Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
5	I37	W	0h	Interrupt associated with TCC #37
4	I36	W	0h	Interrupt associated with TCC #36
3	I35	W	0h	Interrupt associated with TCC #35
2	I34	W	0h	Interrupt associated with TCC #34
1	I33	W	0h	Interrupt associated with TCC #33
0	I32	W	0h	Interrupt associated with TCC #32

### 11.8.1.1.105 IEVAL\_RN Register (Offset = 2078h) [reset = 0h]

IEVAL\_RN is shown in [Figure 11-132](#) and described in [Table 11-137](#).

Return to the [Table 11-32](#).

Interrupt Eval Register

**Figure 11-132. IEVAL\_RN Register**

31	30	29	28	27	26	25	24
RES76							
R-0h							
23	22	21	20	19	18	17	16
RES76							
R-0h							
15	14	13	12	11	10	9	8
RES76							
R-0h							
7	6	5	4	3	2	1	0
RES76						SET	EVAL
R-0h						W-0h	W-0h

**Table 11-137. IEVAL\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RES76	R	0h	RESERVE FIELD
1	SET	W	0h	Interrupt Set: CPU write of '1' to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (IPRn). CPU write of '0' has no effect.
0	EVAL	W	0h	Interrupt Evaluate: CPU write of '1' to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (IPRn). CPU write of '0' has no effect..

### 11.8.1.1.106 QER\_RN Register (Offset = 2080h) [reset = 0h]

QER\_RN is shown in [Figure 11-133](#) and described in [Table 11-138](#).

Return to the [Table 11-32](#).

QDMA Event Register: If QER.En bit is set then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. QER.En bit is cleared when the corresponding event is prioritized and serviced. QER.En is also cleared when user writes a '1' to the QSECR.En bit. If the QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and QEER register is set then the corresponding bit in the QDMA Event Missed Register is set.

**Figure 11-133. QER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES77															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES77								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-138. QER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES77	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.107 QEER\_RN Register (Offset = 2084h) [reset = 0h]

QEER\_RN is shown in [Figure 11-134](#) and described in [Table 11-139](#).

Return to the [Table 11-32](#).

QDMA Event Enable Register: Enabled/disabled QDMA address comparator for QDMA Channel N. QEER.En is not directly writeable. QDMA channels can be enabled via writes to QEESR and can be disabled via writes to QEECR register. QEER.En = 1 The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in QER.En. QEER.En = 0 The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in QER.En.

**Figure 11-134. QEER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES78															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES78								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-139. QEER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES78	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0



### 11.8.1.1.108 QEECR\_RN Register (Offset = 2088h) [reset = 0h]

QEECR\_RN is shown in [Figure 11-135](#) and described in [Table 11-140](#).

Return to the [Table 11-32](#).

QDMA Event Enable Clear Register: CPU write of '1' to the QEECR.En bit causes the QEER.En bit to be cleared. CPU write of '0' has no effect..

**Figure 11-135. QEECR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES79															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES79								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-140. QEECR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES79	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.109 QEESR\_RN Register (Offset = 208Ch) [reset = 0h]

QEESR\_RN is shown in [Figure 11-136](#) and described in [Table 11-141](#).

Return to the [Table 11-32](#).

QDMA Event Enable Set Register: CPU write of '1' to the QEESR.En bit causes the QEESR.En bit to be set. CPU write of '0' has no effect..

**Figure 11-136. QEESR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES80															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES80								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-141. QEESR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES80	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.110 QSER\_RN Register (Offset = 2090h) [reset = 0h]

QSER\_RN is shown in [Figure 11-137](#) and described in [Table 11-142](#).

Return to the [Table 11-32](#).

QDMA Secondary Event Register: The QDMA secondary event register is used along with the QDMA Event Register (QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.

**Figure 11-137. QSER\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES81															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES81								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

**Table 11-142. QSER\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES81	R	0h	RESERVE FIELD
7	E7	R	0h	Event #7
6	E6	R	0h	Event #6
5	E5	R	0h	Event #5
4	E4	R	0h	Event #4
3	E3	R	0h	Event #3
2	E2	R	0h	Event #2
1	E1	R	0h	Event #1
0	E0	R	0h	Event #0

### 11.8.1.1.111 QSECR\_RN Register (Offset = 2094h) [reset = 0h]

QSECR\_RN is shown in [Figure 11-138](#) and described in [Table 11-143](#).

Return to the [Table 11-32](#).

QDMA Secondary Event Clear Register: The secondary event clear register is used to clear the status of the QSER and QER register (note that this is slightly different than the SER operation which does not clear the ER.En register). CPU write of '1' to the QSECR.En bit clears the QSER.En and QER.En register fields. CPU write of '0' has no effect..

**Figure 11-138. QSECR\_RN Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RES82															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES82								E7	E6	E5	E4	E3	E2	E1	E0
R-0h								W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h	W-0h

**Table 11-143. QSECR\_RN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RES82	R	0h	RESERVE FIELD
7	E7	W	0h	Event #7
6	E6	W	0h	Event #6
5	E5	W	0h	Event #5
4	E4	W	0h	Event #4
3	E3	W	0h	Event #3
2	E2	W	0h	Event #2
1	E1	W	0h	Event #1
0	E0	W	0h	Event #0

### 11.8.1.1.112 OPT Register (Offset = 4000h) [reset = 0h]

OPT is shown in [Figure 11-139](#) and described in [Table 11-144](#).

Return to the [Table 11-32](#).

Options Parameter

**Figure 11-139. OPT Register**

31	30	29	28	27	26	25	24
PRIV	RES83			PRIVID			
R-0h	R-0h			R-0h			
23	22	21	20	19	18	17	16
ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	WIMODE	RES84	TCC	
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h	R/W-0h	
15	14	13	12	11	10	9	8
TCC				TCCMODE	FWID		
R/W-0h				R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RES85				STATIC	SYNCDIM	DAM	SAM
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

**Table 11-144. OPT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	PRIV	R	0h	Privilege level: privilege level (supervisor vs. user) for the host/cpu/dma that programmed this PaRAM Entry. Value is set with the vbus priv value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus. PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege
30-28	RES83	R	0h	RESERVE FIELD
27-24	PRIVID	R	0h	Privilege ID: Privilege ID for the external host/cpu/dma that programmed this PaRAM Entry. This value is set with the vbus privid value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus.
23	ITCCHEN	R/W	0h	Intermediate transfer completion chaining enable: 0: Intermediate transfer complete chaining is disabled. 1: Intermediate transfer complete chaining is enabled.
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	ITCINTEN	R/W	0h	Intermediate transfer completion interrupt enable: 0: Intermediate transfer complete interrupt is disabled. 1: Intermediate transfer complete interrupt is enabled (corresponding IER[TCC] bit must be set to 1 to generate interrupt)
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled (corresponding IER[TCC] bit must be set to 1 to generate interrupt)

**Table 11-144. OPT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
19	WIMODE	R/W	0h	Backward compatibility mode: 0: Normal operation 1 : WI Backwards Compatibility mode forces BCNT to be adjusted by '1' upon TR submission (0 means 1 1 means 2 ... ) and forces ACNT to be treated as a word-count (left shifted by 2 by hardware to create byte cnt for TR submission)
18	RES84	R	0h	RESERVE FIELD
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER (bit CER[TCC]) for chaining or in IER (bit IER[TCC]) for interrupts.
11	TCCMODE	R/W	0h	Transfer complete code mode: Indicates the point at which a transfer is considered completed. Applies to both chaining and interrupt. 0: Normal Completion A transfer is considered completed after the transfer parameters are returned to the CC from the TC (which was returned from the peripheral). 1: Early Completion A transfer is considered completed after the CC submits a TR to the TC. CC generates completion code internally .
10-8	FWID	R/W	0h	FIFO width: Applies if either SAM or DAM is set to FIFO mode. Pass-thru to TC.
7-4	RES85	R	0h	RESERVE FIELD
3	STATIC	R/W	0h	Static Entry: 0: Entry is updated as normal 1: Entry is static Count and Address updates are not updated after TRP is submitted. Linking is not performed.
2	SYNCDIM	R/W	0h	Transfer Synchronization Dimension: 0: A-Sync Each event triggers the transfer of ACNT elements. 1: AB-Sync Each event triggers the transfer of BCNT arrays of ACNT elements
1	DAM	R/W	0h	Destination Address Mode: Destination Address Mode within an array. Pass-thru to TC. 0: INCR Dst addressing within an array increments. Dst is not a FIFO. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode: Source Address Mode within an array. Pass-thru to TC. 0: INCR Src addressing within an array increments. Source is not a FIFO. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

### 11.8.1.1.113 SRC Register (Offset = 4004h) [reset = 0h]

SRC is shown in [Figure 11-140](#) and described in [Table 11-145](#).

Return to the [Table 11-32](#).

Source Address

**Figure 11-140. SRC Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC																															
R/W-0h																															

**Table 11-145. SRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SRC	R/W	0h	Source Address: The 32-bit source address parameters specify the starting byte address of the source . If SAM is set to FIFO mode then the user should program the Source address to be aligned to the value specified by the OPT.FWID field. No errors are recognized here but TC will assert error if this is not true.

### 11.8.1.1.114 ABCNT Register (Offset = 4008h) [reset = 0h]

ABCNT is shown in [Figure 11-141](#) and described in [Table 11-146](#).

Return to the [Table 11-32](#).

A and B byte count

**Figure 11-141. ABCNT Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															
R/W-0h																R/W-0h															

**Table 11-146. ABCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	BCNT	R/W	0h	BCNT : Count for 2nd Dimension: BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation valid values for BCNT can be anywhere between 1 and 65535. Therefore the maximum number of arrays in a frame is 65535 (64K-1 arrays). BCNT=1 means 1 array in the frame and BCNT=0 means 0 arrays in the frame. In normal mode a BCNT of '0' is considered as either a Null or Dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the OPT.WIMODE bit is set then the programmed BCNT value will be incremented by '1' before submission to TC. I.e. 0 means 1 1 means 2 2 means 3 ...
15-0	ACNT	R/W	0h	ACNT : number of bytes in 1st dimension: ACNT represents the number of bytes within the first dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65535. Therefore the maximum number of bytes in an array is 65535 bytes (64K-1 bytes). ACNT must be greater than or equal to '1' for a TR to be submitted to TC. An ACNT of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the OPT.WIMODE bit is set then the ACNT field represents a word count. The CC must internally multiply by 4 to translate the word count to a byte count prior to submission to the TC. The 2 MSBs of the 16-bit ACNT are reserved and should always be written as 'b00 by the user. If user writes a value other than 0 it will still be treated as 0 since the multiply-by-4 operation (to translate between a word count and a byte count) will drop the 2 msbits. For dummy and null transfer definition the ACNT definition will disregard the 2 msbits. I.e. a programmed ACNT value of 0x8000 in WI-mode will be treated as 0 byte transfer resulting in null or dummy operation dependent on the state of BCNT and CCNT.



### 11.8.1.1.115 DST Register (Offset = 400Ch) [reset = 0h]

DST is shown in [Figure 11-142](#) and described in [Table 11-147](#).

Return to the [Table 11-32](#).

Destination Address

**Figure 11-142. DST Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST																															
R/W-0h																															

**Table 11-147. DST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DST	R/W	0h	Destination Address: The 32-bit destination address parameters specify the starting byte address of the destination. If DAM is set to FIFO mode then the user should program the Destination address to be aligned to the value specified by the OPT.FWID field. No errors are recognized here but TC will assert error if this is not true.

### 11.8.1.1.116 BIDX Register (Offset = 4010h) [reset = 0h]

BIDX is shown in [Figure 11-143](#) and described in [Table 11-148](#).

Return to the [Table 11-32](#).

Register description is not available

**Figure 11-143. BIDX Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R/W-0h																R/W-0h															

**Table 11-148. BIDX Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	DBIDX	R/W	0h	Destination 2nd Dimension Index: DBIDX is a 16-bit signed value (2's complement) used for destination address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-Sync and AB-Sync transfers.
15-0	SBIDX	R/W	0h	Source 2nd Dimension Index: SBIDX is a 16-bit signed value (2's complement) used for source address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-sync and AB-sync transfers.

### 11.8.1.1.117 LNK Register (Offset = 4014h) [reset = 0h]

LNK is shown in [Figure 11-144](#) and described in [Table 11-149](#).

Return to the [Table 11-32](#).

Link and Reload parameters

**Figure 11-144. LNK Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNTRLD																LINK															
R/W-0h																R/W-0h															

**Table 11-149. LNK Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	BCNTRLD	R/W	0h	BCNT Reload: BCNTRLD is a 16-bit unsigned value used to reload the BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-Sync'ed transfers. In this case the CC decrements the BCNT value by one on each TR submission. When BCNT (conceptually) reaches zero then the CC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value. For AB-synchronized transfers the CC submits the BCNT in the TR and therefore the TC is responsible to keep track of BCNT not thus BCNTRLD is a don't care field.
15-0	LINK	R/W	0h	Link Address: The CC provides a mechanism to reload the current PaRAM Entry upon its natural termination (i.e. after count fields are decremented to '0') with a new PaRAM Entry. This is called 'linking'. The 16-bit parameter LINK specifies the byte address offset in the PaRAM from which the CC loads/reloads the next PaRAM entry in the link. The CC should disregard the value in the upper 2 bits of the LINK field as well as the lower 5-bits of the LINK field. The upper two bits are ignored such that the user can program either the 'literal' byte address of the LINK parameter or the 'PaRAM base-relative' address of the link field. Therefore if the user uses the literal address with a range from 0x4000 to 0x7FFF it will be treated as a PaRAM-base-relative value of 0x0000 to 0x3FFF. The lower-5 bits are ignored and treated as 'b00000 thereby guaranteeing that all Link pointers point to a 32-byte aligned PaRAM entry. In the latter case (5-lsbs) behavior is undefined for the user (i.e. don't have to test it). In the former case (2 msbs) user should be able to take advantage of this feature (i.e. do have to test it). If a Link Update is requested to a PaRAM address that is beyond the actual range of implemented PaRAM then the Link will be treated as a Null Link and all 0s plus 0xFFFF will be written to the current entry location. A LINK value of 0xFFFF is referred to as a NULL link which should cause the CC to write 0x0 to all entries of the current PaRAM Entry except for the LINK field which is set to 0xFFFF. The Priv/Privid/Secure state is overwritten to 0x0 when linking. MSBs and LSBS should not be masked when comparing against the 0xFFFF value. I.e. a value of 0x3FFE is a non-NULL PaRAM link field.

### 11.8.1.1.118 CIDX Register (Offset = 4018h) [reset = 0h]

CIDX is shown in [Figure 11-145](#) and described in [Table 11-150](#).

Return to the [Table 11-32](#).

Register description is not available

**Figure 11-145. CIDX Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCIDX																SCIDX															
R/W-0h																R/W-0h															

**Table 11-150. CIDX Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	DCIDX	R/W	0h	Destination Frame Index: DCIDX is a 16-bit signed value (2's complement) used for destination address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when DCIDX is applied the current array in an A-sync transfer is the last array in the frame while the current array in a ABsync transfer is the first array in the frame.
15-0	SCIDX	R/W	0h	Source Frame Index: SCIDX is a 16-bit signed value (2's complement) used for source address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when SCIDX is applied the current array in an A-sync transfer is the last array in the frame while the current array in a AB-sync transfer is the first array in the frame.

### 11.8.1.1.119 CCNT Register (Offset = 401Ch) [reset = 0h]

CCNT is shown in [Figure 11-146](#) and described in [Table 11-151](#).

Return to the [Table 11-32](#).

C byte count

**Figure 11-146. CCNT Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RES86																CCNT															
R-0h																R/W-0h															

**Table 11-151. CCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RES86	R	0h	RESERVE FIELD
15-0	CCNT	R/W	0h	CCNT : Count for 3rd Dimension: CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT can be anywhere between 1 and 65535. Therefore the maximum number of frames in a block is 65535 (64K-1 frames). CCNT of '1' means '1' frame in the block and CCNT of '0' means '0' frames in the block. A CCNT value of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. WIMODE has no affect on CCNT operation.

### 11.8.1.2 TPTC Registers

Table 11-152 lists the TPTC registers. All register offset addresses not listed in Table 11-152 should be considered as reserved locations and the register contents should not be modified.

**Table 11-152. TPTC Registers**

Offset	Acronym	Register Name	Section
0h	PID	Peripheral ID Register	<a href="#">Section 11.8.1.2.1</a>
4h	TCCFG	TC Configuration Register	<a href="#">Section 11.8.1.2.2</a>
100h	TCSTAT	TC Status Register	<a href="#">Section 11.8.1.2.3</a>
104h	INTSTAT	Interrupt Status Register	<a href="#">Section 11.8.1.2.4</a>
108h	INTEN	Interrupt Enable Register	<a href="#">Section 11.8.1.2.5</a>
10Ch	INTCLR	Interrupt Clear Register	<a href="#">Section 11.8.1.2.6</a>
110h	INTCMD	Interrupt Command Register	<a href="#">Section 11.8.1.2.7</a>
120h	ERRSTAT	Error Status Register	<a href="#">Section 11.8.1.2.8</a>
124h	ERREN	Error Enable Register	<a href="#">Section 11.8.1.2.9</a>
128h	ERRCLR	Error Clear Register	<a href="#">Section 11.8.1.2.10</a>
12Ch	ERRDET	Error Details Register	<a href="#">Section 11.8.1.2.11</a>
130h	ERRCMD	Error Command Register	<a href="#">Section 11.8.1.2.12</a>
140h	RDRATE	Read Rate Register	<a href="#">Section 11.8.1.2.13</a>
200h	POPT	Prog Set Options	<a href="#">Section 11.8.1.2.14</a>
204h	PSRC	Prog Set Src Address	<a href="#">Section 11.8.1.2.15</a>
208h	PCNT	Prog Set Count	<a href="#">Section 11.8.1.2.16</a>
20Ch	PDST	Prog Set Dst Address	<a href="#">Section 11.8.1.2.17</a>
210h	PBIDX	Prog Set B-Dim Idx	<a href="#">Section 11.8.1.2.18</a>
214h	PMPPRXY	Prog Set Mem Protect Proxy	<a href="#">Section 11.8.1.2.19</a>
240h	SAOPT	Src Actv Set Options	<a href="#">Section 11.8.1.2.20</a>
244h	SASRC	Src Actv Set Src Address	<a href="#">Section 11.8.1.2.21</a>
248h	SACNT	Src Actv Set A-Count	<a href="#">Section 11.8.1.2.22</a>
24Ch	SADST	Src Actv Set Dst Address	<a href="#">Section 11.8.1.2.23</a>
250h	SABIDX	Src Actv Set B-Dim Idx	<a href="#">Section 11.8.1.2.24</a>
254h	SAMPPRXY	Src Actv Set Mem Protect Proxy	<a href="#">Section 11.8.1.2.25</a>
258h	SACNTRLD	Src Actv Set Cnt Reload	<a href="#">Section 11.8.1.2.26</a>
25Ch	SASRCBREF	Src Actv Set Src Addr B-Reference	<a href="#">Section 11.8.1.2.27</a>
260h	SADSTBREF	Src Actv Set Dst Addr B-Reference	<a href="#">Section 11.8.1.2.28</a>
264h	SABCNT	Src Actv Set B-Count	<a href="#">Section 11.8.1.2.29</a>
280h	DFCNTRLD	Dst FIFO Set Cnt Reload	<a href="#">Section 11.8.1.2.30</a>
284h	DFSRCBREF	Dst FIFO Set Src Addr B-Reference	<a href="#">Section 11.8.1.2.31</a>
300h	DFOPT0	Dst FIFO Set Options	<a href="#">Section 11.8.1.2.32</a>
304h	DFSRC0	Dst FIFO Set Src Address	<a href="#">Section 11.8.1.2.33</a>
308h	DFACNT0	Dst FIFO Set A-Count	<a href="#">Section 11.8.1.2.34</a>
30Ch	DFDST0	Dst FIFO Set Dst Address	<a href="#">Section 11.8.1.2.35</a>
310h	DFBIDX0	Dst FIFO Set B-Dim Idx	<a href="#">Section 11.8.1.2.36</a>
314h	DFMPPRXY0	Dst FIFO Set Mem Protect Proxy	<a href="#">Section 11.8.1.2.37</a>
318h	DFBCNT0	Dst FIFO Set B-Count	<a href="#">Section 11.8.1.2.38</a>
340h	DFOPT1	Dst FIFO Set Options	<a href="#">Section 11.8.1.2.39</a>
344h	DFSRC1	Dst FIFO Set Src Address	<a href="#">Section 11.8.1.2.40</a>
348h	DFACNT1	Dst FIFO Set A-Count	<a href="#">Section 11.8.1.2.41</a>

**Table 11-152. TPTC Registers (continued)**

Offset	Acronym	Register Name	Section
34Ch	DFDST1	Dst FIFO Set Dst Address	<a href="#">Section 11.8.1.2.42</a>
350h	DFBIDX1	Dst FIFO Set B-Dim Idx	<a href="#">Section 11.8.1.2.43</a>
354h	DFMPPRXY1	Dst FIFO Set Mem Protect Proxy	<a href="#">Section 11.8.1.2.44</a>
358h	DFBCNT1	Dst FIFO Set B-Count	<a href="#">Section 11.8.1.2.45</a>

### 11.8.1.2.1 PID Register (Offset = 0h) [reset = X]

PID is shown in [Figure 11-147](#) and described in [Table 11-153](#).

Return to the [Table 11-152](#).

Peripheral ID Register

**Figure 11-147. PID Register**

31	30	29	28	27	26	25	24
SCHEME		RESERVED		FUNC			
R-1h		R-X		R-0h			
23	22	21	20	19	18	17	16
FUNC							
R-0h							
15	14	13	12	11	10	9	8
RTL					MAJOR		
R-1h					R-3h		
7	6	5	4	3	2	1	0
CUSTOM		MINOR					
R-0h		R-1h					

**Table 11-153. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID Scheme: Used to distinguish between old ID scheme and current. Spare bit to encode future schemes EDMA uses 'new scheme' indicated with value of 0x1.
29-28	RESERVED	R	X	
27-16	FUNC	R	0h	Function indicates a software compatible module family.
15-11	RTL	R	1h	RTL Version
10-8	MAJOR	R	3h	Major Revision
7-6	CUSTOM	R	0h	Custom revision field: Not used on this version of EDMA.
5-0	MINOR	R	1h	Minor Revision



### 11.8.1.2.2 TCCFG Register (Offset = 4h) [reset = X]

TCCFG is shown in [Figure 11-148](#) and described in [Table 11-154](#).

Return to the [Table 11-152](#).

TC Configuration Register

**Figure 11-148. TCCFG Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED						DREGDEPTH	
R-X						R-2h	
7	6	5	4	3	2	1	0
RESERVED		BUSWIDTH		RESERVED		FIFOSIZE	
R-X		R-2h		R-X		R-4h	

**Table 11-154. TCCFG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9-8	DREGDEPTH	R	2h	Dst Register FIFO Depth Parameterization
7-6	RESERVED	R	X	
5-4	BUSWIDTH	R	2h	Bus Width Parameterization
3	RESERVED	R	X	
2-0	FIFOSIZE	R	4h	Fifo Size Parameterization

### 11.8.1.2.3 TCSTAT Register (Offset = 100h) [reset = X]

TCSTAT is shown in [Figure 11-149](#) and described in [Table 11-155](#).

Return to the [Table 11-152](#).

TC Status Register

**Figure 11-149. TCSTAT Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED		DFSTRTPTR		RESERVED		ACTV	
R-X		R-0h		R-X		R-1h	
7	6	5	4	3	2	1	0
RESERVED	DSTACTV			RESERVED	WSACTV	SRCACTV	PROGBUSY
R-X	R-0h			R-X	R-0h	R-0h	R-0h

**Table 11-155. TCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-14	RESERVED	R	X	
13-12	DFSTRTPTR	R	0h	Dst FIFO Start Pointer Represents the offset to the head entry of Dst Register FIFO in units of entries. Legal values = 0x0 to 0x3
11-9	RESERVED	R	X	
8	ACTV	R	1h	Channel Active Channel Active is a logical-OR of each of the BUSY/ACTV signals. The ACTV bit must remain high through the life of a TR. ACTV = 0 : Channel is idle. ACTV = 1 : Channel is busy.
7	RESERVED	R	X	
6-4	DSTACTV	R	0h	Destination Active State Specifies the number of TRs that are resident in the Dst Register FIFO at a given instant. Legal values are constrained by the DSTREGDEPTH parameter.
3	RESERVED	R	X	
2	WSACTV	R	0h	Write Status Active WSACTV = 0 : Write status is not pending. Write status has been received for all previously issued write commands. WSACTV = 1 : Write Status is pending. Write status has not been received for all previously issued write commands.

**Table 11-155. TCSTAT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	SRCACTV	R	0h	Source Active State SRCACTV = 0 : Source Active set is idle. Any TR written to Prog Set will immediately transition to Source Active set as long as the Dst FIFO Set is not full [DSTFULL == 1]. SRCACTV = 1 : Source Active set is busy either performing read transfers or waiting to perform read transfers for current Transfer Request.
0	PROGBUSY	R	0h	Program Register Set Busy PROGBUSY = 0 : Prog set idle and is available for programming. PROGBUSY = 1 : Prog set busy. User should poll for PROGBUSY equal to '0' prior to re-programming the Program Register set.

#### 11.8.1.2.4 INTSTAT Register (Offset = 104h) [reset = X]

INTSTAT is shown in [Figure 11-150](#) and described in [Table 11-156](#).

Return to the [Table 11-152](#).

Interrupt Status Register

**Figure 11-150. INTSTAT Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED							
R-X							
7	6	5	4	3	2	1	0
RESERVED						TRDONE	PROGEMPTY
R-X						R-0h	R-0h

**Table 11-156. INTSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R	X	
1	TRDONE	R	0h	TR Done Event Status: TRDONE = 0 : Condition not detected. TRDONE = 1 : Set when TC has completed a Transfer Request. TRDONE should be set when the write status is returned for the final write of a TR. Cleared when user writes '1' to INTCLR.TRDONE register bit.
0	PROGEMPTY	R	0h	Program Set Empty Event Status: PROGEMPTY = 0 : Condition not detected. PROGEMPTY = 1 : Set when Program Register set transitions to empty state. Cleared when user writes '1' to INTCLR.PROGEMPTY register bit.

### 11.8.1.2.5 INTEN Register (Offset = 108h) [reset = X]

INTEN is shown in [Figure 11-151](#) and described in [Table 11-157](#).

Return to the [Table 11-152](#).

Interrupt Enable Register

**Figure 11-151. INTEN Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						TRDONE	PROGEMPTY
R/W-X						R/W-0h	R/W-0h

**Table 11-157. INTEN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	TRDONE	R/W	0h	TR Done Event Enable: INTEN.TRDONE = 0 : TRDONE Event is disabled. INTEN.TRDONE = 1 : TRDONE Event is enabled and contributes to interrupt generation
0	PROGEMPTY	R/W	0h	Program Set Empty Event Enable: INTEN.PROGEMPTY = 0 : PROGEMPTY Event is disabled. INTEN.PROGEMPTY = 1 : PROGEMPTY Event is enabled and contributes to interrupt generation

### 11.8.1.2.6 INTCLR Register (Offset = 10Ch) [reset = X]

INTCLR is shown in [Figure 11-152](#) and described in [Table 11-158](#).

Return to the [Table 11-152](#).

Interrupt Clear Register

**Figure 11-152. INTCLR Register**

31	30	29	28	27	26	25	24
RESERVED							
W-X							
23	22	21	20	19	18	17	16
RESERVED							
W-X							
15	14	13	12	11	10	9	8
RESERVED							
W-X							
7	6	5	4	3	2	1	0
RESERVED						TRDONE	PROGEMPTY
W-X						W-0h	W-0h

**Table 11-158. INTCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	X	
1	TRDONE	W	0h	TR Done Event Clear: INTCLR.TRDONE = 0 : Writes of '0' have no effect. INTCLR.TRDONE = 1 : Write of '1' clears INTSTAT.TRDONE bit
0	PROGEMPTY	W	0h	Program Set Empty Event Clear: INTCLR.PROGEMPTY = 0 : Writes of '0' have no effect. INTCLR.PROGEMPTY = 1 : Write of '1' clears INTSTAT.PROGEMPTY bit

### 11.8.1.2.7 INTCMD Register (Offset = 110h) [reset = X]

INTCMD is shown in [Figure 11-153](#) and described in [Table 11-159](#).

Return to the [Table 11-152](#).

Interrupt Command Register

**Figure 11-153. INTCMD Register**

31	30	29	28	27	26	25	24
RESERVED							
W-X							
23	22	21	20	19	18	17	16
RESERVED							
W-X							
15	14	13	12	11	10	9	8
RESERVED							
W-X							
7	6	5	4	3	2	1	0
RESERVED						SET	EVAL
W-X						W-0h	W-0h

**Table 11-159. INTCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	X	
1	SET	W	0h	Set TPTC interrupt: Write of '1' to SET causes TPTC interrupt to be pulsed unconditionally. Writes of '0' have no affect.
0	EVAL	W	0h	Evaluate state of TPTC interrupt Write of '1' to EVAL causes TPTC interrupt to be pulsed if any of the INTSTAT bits are set to '1'. Writes of '0' have no affect.

### 11.8.1.2.8 ERRSTAT Register (Offset = 120h) [reset = X]

ERRSTAT is shown in [Figure 11-154](#) and described in [Table 11-160](#).

Return to the [Table 11-152](#).

Error Status Register

**Figure 11-154. ERRSTAT Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED							
R-X							
7	6	5	4	3	2	1	0
RESERVED				MMRAERR	TRERR	RESERVED	BUSERR
R-X				R-0h	R-0h	R-X	R-0h

**Table 11-160. ERRSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	X	
3	MMRAERR	R	0h	MMR Address Error: MMRAERR = 0 : Condition not detected. MMRAERR = 1 : User attempted to read or write to invalid address configuration memory map. [Is only be set for non-emulation accesses]. No additional error information is recorded.
2	TRERR	R	0h	TR Error: TR detected that violates FIFO Mode transfer [SAM or DAM is '1'] alignment rules or has ACNT or BCNT == 0. No additional error information is recorded.
1	RESERVED	R	X	
0	BUSERR	R	0h	Bus Error Event: BUSERR = 0: Condition not detected. BUSERR = 1: TC has detected an error code on the write response bus or read response bus. Error information is stored in Error Details Register [ERRDET].



### 11.8.1.2.9 ERREN Register (Offset = 124h) [reset = X]

ERREN is shown in [Figure 11-155](#) and described in [Table 11-161](#).

Return to the [Table 11-152](#).

Error Enable Register

**Figure 11-155. ERREN Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				MMRAERR	TRERR	RESERVED	BUSERR
R/W-X				R/W-0h	R/W-0h	R/W-X	R/W-0h

**Table 11-161. ERREN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3	MMRAERR	R/W	0h	Interrupt enable for ERRSTAT.MMRAERR: ERREN.MMRAERR = 0 : BUSERR is disabled. ERREN.MMRAERR = 1 : MMRAERR is enabled and contributes to the TPTC error interrupt generation.
2	TRERR	R/W	0h	Interrupt enable for ERRSTAT.TRERR: ERREN.TRERR = 0 : BUSERR is disabled. ERREN.TRERR = 1 : TRERR is enabled and contributes to the TPTC error interrupt generation.
1	RESERVED	R/W	X	
0	BUSERR	R/W	0h	Interrupt enable for ERRSTAT.BUSERR: ERREN.BUSERR = 0 : BUSERR is disabled. ERREN.BUSERR = 1 : BUSERR is enabled and contributes to the TPTC error interrupt generation.

### 11.8.1.2.10 ERRCLR Register (Offset = 128h) [reset = X]

ERRCLR is shown in [Figure 11-156](#) and described in [Table 11-162](#).

Return to the [Table 11-152](#).

Error Clear Register

**Figure 11-156. ERRCLR Register**

31	30	29	28	27	26	25	24
RESERVED							
W-X							
23	22	21	20	19	18	17	16
RESERVED							
W-X							
15	14	13	12	11	10	9	8
RESERVED							
W-X							
7	6	5	4	3	2	1	0
RESERVED				MMRAERR	TRERR	RESERVED	BUSERR
W-X				W-0h	W-0h	W-X	W-0h

**Table 11-162. ERRCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	W	X	
3	MMRAERR	W	0h	Interrupt clear for ERRSTAT.MMRAERR: ERRCLR.MMRAERR = 0 : Writes of '0' have no effect. ERRCLR.MMRAERR = 1 : Write of '1' clears ERRSTAT.MMRAERR bit. Write of '1' to ERRCLR.MMRAERR does not clear the ERRDET register.
2	TRERR	W	0h	Interrupt clear for ERRSTAT.TRERR: ERRCLR.TRERR = 0 : Writes of '0' have no effect. ERRCLR.TRERR = 1 : Write of '1' clears ERRSTAT.TRERR bit. Write of '1' to ERRCLR.TRERR does not clear the ERRDET register.
1	RESERVED	W	X	
0	BUSERR	W	0h	Interrupt clear for ERRSTAT.BUSERR: ERRCLR.BUSERR = 0 : Writes of '0' have no effect. ERRCLR.BUSERR = 1 : Write of '1' clears ERRSTAT.BUSERR bit. Write of '1' to ERRCLR.BUSERR clears the ERRDET register.

### 11.8.1.2.11 ERRDET Register (Offset = 12Ch) [reset = X]

ERRDET is shown in [Figure 11-157](#) and described in [Table 11-163](#).

Return to the [Table 11-152](#).

Error Details Register

**Figure 11-157. ERRDET Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED						TCCHEN	TCINTEN
R-X						R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			TCC				
R-X			R-0h				
7	6	5	4	3	2	1	0
RESERVED				STAT			
R-X				R-0h			

**Table 11-163. ERRDET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-18	RESERVED	R	X	
17	TCCHEN	R	0h	Contains the OPT.TCCHEN value programmed by the user for the Read or Write transaction that resulted in an error.
16	TCINTEN	R	0h	Contains the OPT.TCINTEN value programmed by the user for the Read or Write transaction that resulted in an error.
15-14	RESERVED	R	X	
13-8	TCC	R	0h	Transfer Complete Code: Contains the OPT.TCC value programmed by the user for the Read or Write transaction that resulted in an error.
7-4	RESERVED	R	X	
3-0	STAT	R	0h	Transaction Status: Stores the non-zero status/error code that was detected on the read status or write status bus. MS-bit effectively serves as the read vs. write error code. If read status and write status are returned on the same cycle then the TC chooses non-zero version. If both are non-zero then write status is treated as higher priority. Encoding of errors matches the CBA spec.

### 11.8.1.2.12 ERRCMD Register (Offset = 130h) [reset = X]

ERRCMD is shown in [Figure 11-158](#) and described in [Table 11-164](#).

Return to the [Table 11-152](#).

Error Command Register

**Figure 11-158. ERRCMD Register**

31	30	29	28	27	26	25	24
RESERVED							
W-X							
23	22	21	20	19	18	17	16
RESERVED							
W-X							
15	14	13	12	11	10	9	8
RESERVED							
W-X							
7	6	5	4	3	2	1	0
RESERVED						SET	EVAL
W-X						W-0h	W-0h

**Table 11-164. ERRCMD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	W	X	
1	SET	W	0h	Set TPTC error interrupt: Write of '1' to SET causes TPTC error interrupt to be pulsed unconditionally. Writes of '0' have no affect.
0	EVAL	W	0h	Evaluate state of TPTC error interrupt Write of '1' to EVAL causes TPTC error interrupt to be pulsed if any of the ERRSTAT bits are set to '1'. Writes of '0' have no affect.

### 11.8.1.2.13 RDRATE Register (Offset = 140h) [reset = X]

RDRATE is shown in [Figure 11-159](#) and described in [Table 11-165](#).

Return to the [Table 11-152](#).

Read Rate Register

**Figure 11-159. RDRATE Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
RESERVED															
R/W-X															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RDRATE			
R/W-X												R/W-0h			

**Table 11-165. RDRATE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R/W	X	
2-0	RDRATE	R/W	0h	Read Rate Control: Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this TC.

#### 11.8.1.2.14 POPT Register (Offset = 200h) [reset = X]

POPT is shown in [Figure 11-160](#) and described in [Table 11-166](#).

Return to the [Table 11-152](#).

Prog Set Options

**Figure 11-160. POPT Register**

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

**Table 11-166. POPT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tpc_r_dbg_channel_id) and write (tpc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

**Table 11-166. POPT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	DAM	R/W	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

### 11.8.1.2.15 PSRC Register (Offset = 204h) [reset = 0h]

PSRC is shown in [Figure 11-161](#) and described in [Table 11-167](#).

Return to the [Table 11-152](#).

Prog Set Src Address

**Figure 11-161. PSRC Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R/W-0h																															

**Table 11-167. PSRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SADDR	R/W	0h	Source address for Program Register Set



### 11.8.1.2.16 PCNT Register (Offset = 208h) [reset = 0h]

PCNT is shown in [Figure 11-162](#) and described in [Table 11-168](#).

Return to the [Table 11-152](#).

Prog Set Count

**Figure 11-162. PCNT Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															
R/W-0h																R/W-0h															

**Table 11-168. PCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	BCNT	R/W	0h	B-Dimension count. Number of arrays to be transferred where each array is ACNT in length.
15-0	ACNT	R/W	0h	A-Dimension count. Number of bytes to be transferred in first dimension.

### 11.8.1.2.17 PDST Register (Offset = 20Ch) [reset = 0h]

PDST is shown in [Figure 11-163](#) and described in [Table 11-169](#).

Return to the [Table 11-152](#).

Prog Set Dst Address

**Figure 11-163. PDST Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															
R/W-0h																															

**Table 11-169. PDST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DADDR	R/W	0h	Destination address for Program Register Set

### 11.8.1.2.18 PBIDX Register (Offset = 210h) [reset = 0h]

PBIDX is shown in [Figure 11-164](#) and described in [Table 11-170](#).

Return to the [Table 11-152](#).

Prog Set B-Dim Idx

**Figure 11-164. PBIDX Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R/W-0h																R/W-0h															

**Table 11-170. PBIDX Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	DBIDX	R/W	0h	Dest B-Idx for Program Register Set: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array [recall that there are BCNT arrays of ACNT elements]. DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R/W	0h	Source B-Idx for Program Register Set: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array [recall that there are BCNT arrays of ACNT elements]. SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

### 11.8.1.2.19 PMPPRXY Register (Offset = 214h) [reset = X]

PMPPRXY is shown in [Figure 11-165](#) and described in [Table 11-171](#).

Return to the [Table 11-152](#).

Prog Set Mem Protect Proxy

**Figure 11-165. PMPPRXY Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED						SECURE	PRIV
R-X						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				PRIVID			
R-X				R-0h			

**Table 11-171. PMPPRXY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register [trigger register]. The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RESERVED	R	X	
3-0	PRIVID	R	0h	Privilege ID: PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register [trigger register]. The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.

### 11.8.1.2.20 SAOPT Register (Offset = 240h) [reset = X]

SAOPT is shown in [Figure 11-166](#) and described in [Table 11-172](#).

Return to the [Table 11-152](#).

Src Actv Set Options

**Figure 11-166. SAOPT Register**

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

**Table 11-172. SAOPT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tpc_r_dbg_channel_id) and write (tpc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

**Table 11-172. SAOPT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	DAM	R/W	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

### 11.8.1.2.21 SASRC Register (Offset = 244h) [reset = 0h]

SASRC is shown in [Figure 11-167](#) and described in [Table 11-173](#).

Return to the [Table 11-152](#).

Src Actv Set Src Address

**Figure 11-167. SASRC Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R-0h																															

**Table 11-173. SASRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address for Source Active Register Set

### 11.8.1.2.22 SACNT Register (Offset = 248h) [reset = X]

SACNT is shown in [Figure 11-168](#) and described in [Table 11-174](#).

Return to the [Table 11-152](#).

Src Actv Set A-Count

**Figure 11-168. SACNT Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										ACNT																					
R-X										R-0h																					

**Table 11-174. SACNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-0	ACNT	R	0h	A-Dimension count. Number of bytes to be transferred in first dimension.



### 11.8.1.2.23 SADST Register (Offset = 24Ch) [reset = 0h]

SADST is shown in [Figure 11-169](#) and described in [Table 11-175](#).

Return to the [Table 11-152](#).

Src Actv Set Dst Address

**Figure 11-169. SADST Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															
R-0h																															

**Table 11-175. SADST Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	Destination address for Source Active Register Set

### 11.8.1.2.24 SABIDX Register (Offset = 250h) [reset = 0h]

SABIDX is shown in [Figure 11-170](#) and described in [Table 11-176](#).

Return to the [Table 11-152](#).

Src Actv Set B-Dim Idx

**Figure 11-170. SABIDX Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R-0h																R-0h															

**Table 11-176. SABIDX Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	Dest B-Idx for Source Active Register Set: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array [recall that there are BCNT arrays of ACNT elements]. DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R	0h	Source B-Idx for Source Active Register Set: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array [recall that there are BCNT arrays of ACNT elements]. SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

### 11.8.1.2.25 SAMPPRXY Register (Offset = 254h) [reset = X]

SAMPPRXY is shown in [Figure 11-171](#) and described in [Table 11-177](#).

Return to the [Table 11-152](#).

Src Actv Set Mem Protect Proxy

**Figure 11-171. SAMPPRXY Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED						SECURE	PRIV
R-X						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				PRIVID			
R-X				R-0h			

**Table 11-177. SAMPPRXY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register [trigger register]. The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RESERVED	R	X	
3-0	PRIVID	R	0h	Privilege ID: PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register [trigger register]. The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.

### 11.8.1.2.26 SACNTRLD Register (Offset = 258h) [reset = X]

SACNTRLD is shown in [Figure 11-172](#) and described in [Table 11-178](#).

Return to the [Table 11-152](#).

Src Actv Set Cnt Reload

**Figure 11-172. SACNTRLD Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															
R-X																R-0h															

**Table 11-178. SACNTRLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	ACNTRLD	R	0h	<p>A-Cnt Reload value for Source Active Register set. Value copied from PCNT.ACNT: Represents the originally programmed value of ACNT.</p> <p>The Reload value is used to reinitialize ACNT after each array is serviced [i.e. ACNT decrements to 0]. by the Src offset in bytes between the starting address of each source array [recall that there are BCNT arrays of ACNT bytes]</p>

### 11.8.1.2.27 SASRCBREF Register (Offset = 25Ch) [reset = 0h]

SASRCBREF is shown in [Figure 11-173](#) and described in [Table 11-179](#).

Return to the [Table 11-152](#).

Src Actv Set Src Addr B-Reference

**Figure 11-173. SASRCBREF Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															
R-0h																															

**Table 11-179. SASRCBREF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SADDRBREF	R	0h	Source address reference for Source Active Register Set: Represents the starting address for the array currently being read. The next array's starting address is calculated as the 'reference address' plus the 'source b-idx' value.

### 11.8.1.2.28 SADSTBREF Register (Offset = 260h) [reset = 0h]

SADSTBREF is shown in [Figure 11-174](#) and described in [Table 11-180](#).

Return to the [Table 11-152](#).

Src Actv Set Dst Addr B-Reference

**Figure 11-174. SADSTBREF Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															
R-0h																															

**Table 11-180. SADSTBREF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DADDRBREF	R	0h	Dst address reference is not applicable for Src Active Register Set. Reads return 0x0.

### 11.8.1.2.29 SABCNT Register (Offset = 264h) [reset = X]

SABCNT is shown in [Figure 11-175](#) and described in [Table 11-181](#).

Return to the [Table 11-152](#).

Src Actv Set B-Count

**Figure 11-175. SABCNT Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCNT															
R-X																R-0h															

**Table 11-181. SABCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	BCNT	R	0h	<p>B-Dimension count:</p> <p>Number of arrays to be transferred where each array is ACNT in length.</p> <p>Count Remaining for Src Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.</p>

### 11.8.1.2.30 DFCNTRLD Register (Offset = 280h) [reset = X]

DFCNTRLD is shown in [Figure 11-176](#) and described in [Table 11-182](#).

Return to the [Table 11-152](#).

Dst FIFO Set Cnt Reload

**Figure 11-176. DFCNTRLD Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															
R-X																R-0h															

**Table 11-182. DFCNTRLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	ACNTRLD	R	0h	<p>A-Cnt Reload value for Destination FIFO Register set. Value copied from PCNT.ACNT: Represents the originally programmed value of ACNT.</p> <p>The Reload value is used to reinitialize ACNT after each array is serviced [i.e. ACNT decrements to 0]. by the Src offset in bytes between the starting address of each source array [recall that there are BCNT arrays of ACNT bytes]</p>



### 11.8.1.2.31 DFSRCBREF Register (Offset = 284h) [reset = 0h]

DFSRCBREF is shown in [Figure 11-177](#) and described in [Table 11-183](#).

Return to the [Table 11-152](#).

Dst FIFO Set Src Addr B-Reference

**Figure 11-177. DFSRCBREF Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															
R-0h																															

**Table 11-183. DFSRCBREF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SADDRBREF	R	0h	Source address reference for Destination FIFO Register Set: Represents the starting address for the array currently being read. The next array's starting address is calculated as the 'reference address' plus the 'source b-idx' value.

### 11.8.1.2.32 DFOPT0 Register (Offset = 300h) [reset = X]

DFOPT0 is shown in [Figure 11-178](#) and described in [Table 11-184](#).

Return to the [Table 11-152](#).

Dst FIFO Set Options

**Figure 11-178. DFOPT0 Register**

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

**Table 11-184. DFOPT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tpc_r_dbg_channel_id) and write (tpc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

**Table 11-184. DFOPT0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	DAM	R/W	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

### 11.8.1.2.33 DFSRC0 Register (Offset = 304h) [reset = 0h]

DFSRC0 is shown in [Figure 11-179](#) and described in [Table 11-185](#).

Return to the [Table 11-152](#).

Dst FIFO Set Src Address

**Figure 11-179. DFSRC0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R-0h																															

**Table 11-185. DFSRC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address is not applicable for Dst FIFO Register Set: Reads return 0x0.

### 11.8.1.2.34 DFACNT0 Register (Offset = 308h) [reset = X]

DFACNT0 is shown in [Figure 11-180](#) and described in [Table 11-186](#).

Return to the [Table 11-152](#).

Dst FIFO Set A-Count

**Figure 11-180. DFACNT0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										ACNT																					
R-X										R-0h																					

**Table 11-186. DFACNT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-0	ACNT	R	0h	A-Dimension count. Number of bytes to be transferred in first dimension.

### 11.8.1.2.35 DFDST0 Register (Offset = 30Ch) [reset = 0h]

DFDST0 is shown in [Figure 11-181](#) and described in [Table 11-187](#).

Return to the [Table 11-152](#).

Dst FIFO Set Dst Address

**Figure 11-181. DFDST0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															
R-0h																															

**Table 11-187. DFDST0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	<p>Destination address for Dst FIFO Register Set: Initial value is copied from PDST.DADDR.</p> <p>TC updates value according to destination addressing mode [OPT.SAM] and/or dest index value [BIDX.DBIDX] after each write command is issued.</p> <p>When a TR is complete the final value should be the address of the last write command issued.</p>

### 11.8.1.2.36 DFBIDX0 Register (Offset = 310h) [reset = 0h]

DFBIDX0 is shown in [Figure 11-182](#) and described in [Table 11-188](#).

Return to the [Table 11-152](#).

Dst FIFO Set B-Dim Idx

**Figure 11-182. DFBIDX0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R-0h																R-0h															

**Table 11-188. DFBIDX0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	Dest B-Idx for Dest FIFO Register Set. Value copied from PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array [recall that there are BCNT arrays of ACNT elements]. DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R	0h	Src B-Idx for Dest FIFO Register Set. Value copied from PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array [recall that there are BCNT arrays of ACNT elements]. SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

### 11.8.1.2.37 DFMPPRXY0 Register (Offset = 314h) [reset = X]

DFMPPRXY0 is shown in [Figure 11-183](#) and described in [Table 11-189](#).

Return to the [Table 11-152](#).

Dst FIFO Set Mem Protect Proxy

**Figure 11-183. DFMPPRXY0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED						SECURE	PRIV
R-X						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				PRIVID			
R-X				R-0h			

**Table 11-189. DFMPPRXY0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register [trigger register]. The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RESERVED	R	X	
3-0	PRIVID	R	0h	Privilege ID: PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register [trigger register]. The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.



### 11.8.1.2.38 DFBCNT0 Register (Offset = 318h) [reset = X]

DFBCNT0 is shown in [Figure 11-184](#) and described in [Table 11-190](#).

Return to the [Table 11-152](#).

Dst FIFO Set B-Count

**Figure 11-184. DFBCNT0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCNT															
R-X																R-0h															

**Table 11-190. DFBCNT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	BCNT	R	0h	<p>B-Count Remaining for Dst Register Set:</p> <p>Number of arrays to be transferred where each array is ACNT in length.</p> <p>Represents the amount of data remaining to be written.</p> <p>Initial value is copied from PCNT.</p> <p>TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.</p>

### 11.8.1.2.39 DFOPT1 Register (Offset = 340h) [reset = X]

DFOPT1 is shown in [Figure 11-185](#) and described in [Table 11-191](#).

Return to the [Table 11-152](#).

Dst FIFO Set Options

**Figure 11-185. DFOPT1 Register**

31	30	29	28	27	26	25	24
RESERVED		DBG_ID		RESERVED			
R/W-X		R/W-0h		R/W-X			
23	22	21	20	19	18	17	16
RESERVED	TCCHEN	RESERVED	TCINTEN	RESERVED		TCC	
R/W-X	R/W-0h	R/W-X	R/W-0h	R/W-X		R/W-0h	
15	14	13	12	11	10	9	8
TCC				RESERVED	FWID		
R/W-0h				R/W-X		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED	PRI			RESERVED		DAM	SAM
R/W-X	R/W-0h			R/W-X		R/W-0h	R/W-0h

**Table 11-191. DFOPT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	RESERVED	R/W	X	
29-28	DBG_ID	R/W	0h	Debug ID Value driven on the read (tpc_r_dbg_channel_id) and write (tpc_w_dbg_channel_id) command bus. Used at system level for trace/profiling of user selected transfers in systems that include this feature.
27-23	RESERVED	R/W	X	
22	TCCHEN	R/W	0h	Transfer complete chaining enable: 0: Transfer complete chaining is disabled. 1: Transfer complete chaining is enabled.
21	RESERVED	R/W	X	
20	TCINTEN	R/W	0h	Transfer complete interrupt enable: 0: Transfer complete interrupt is disabled. 1: Transfer complete interrupt is enabled.
19-18	RESERVED	R/W	X	
17-12	TCC	R/W	0h	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or IPR of the TPCC module.
11	RESERVED	R/W	X	
10-8	FWID	R/W	0h	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.
7	RESERVED	R/W	X	
6-4	PRI	R/W	0h	Transfer Priority: 0: Priority 0 - Highest priority 1: Priority 1 ... 7: Priority 7 - Lowest priority
3-2	RESERVED	R/W	X	

**Table 11-191. DFOPT1 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	DAM	R/W	0h	Destination Address Mode within an array: 0: INCR Dst addressing within an array increments. 1: FIFO Dst addressing within an array wraps around upon reaching FIFO width.
0	SAM	R/W	0h	Source Address Mode within an array: 0: INCR Src addressing within an array increments. 1: FIFO Src addressing within an array wraps around upon reaching FIFO width.

#### 11.8.1.2.40 DFSRC1 Register (Offset = 344h) [reset = 0h]

DFSRC1 is shown in [Figure 11-186](#) and described in [Table 11-192](#).

Return to the [Table 11-152](#).

Dst FIFO Set Src Address

**Figure 11-186. DFSRC1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															
R-0h																															

**Table 11-192. DFSRC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SADDR	R	0h	Source address is not applicable for Dst FIFO Register Set: Reads return 0x0.

#### 11.8.1.2.41 DFACNT1 Register (Offset = 348h) [reset = X]

DFACNT1 is shown in [Figure 11-187](#) and described in [Table 11-193](#).

Return to the [Table 11-152](#).

Dst FIFO Set A-Count

**Figure 11-187. DFACNT1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									ACNT																						
R-X									R-0h																						

**Table 11-193. DFACNT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R	X	
22-0	ACNT	R	0h	A-Dimension count. Number of bytes to be transferred in first dimension.

#### 11.8.1.2.42 DFDST1 Register (Offset = 34Ch) [reset = 0h]

DFDST1 is shown in [Figure 11-188](#) and described in [Table 11-194](#).

Return to the [Table 11-152](#).

Dst FIFO Set Dst Address

**Figure 11-188. DFDST1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															
R-0h																															

**Table 11-194. DFDST1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DADDR	R	0h	<p>Destination address for Dst FIFO Register Set: Initial value is copied from PDST.DADDR.</p> <p>TC updates value according to destination addressing mode [OPT.SAM] and/or dest index value [BIDX.DBIDX] after each write command is issued.</p> <p>When a TR is complete the final value should be the address of the last write command issued.</p>

### 11.8.1.2.43 DFBIDX1 Register (Offset = 350h) [reset = 0h]

DFBIDX1 is shown in [Figure 11-189](#) and described in [Table 11-195](#).

Return to the [Table 11-152](#).

Dst FIFO Set B-Dim Idx

**Figure 11-189. DFBIDX1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															
R-0h																R-0h															

**Table 11-195. DFBIDX1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	DBIDX	R	0h	Dest B-Idx for Dest FIFO Register Set. Value copied from PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array [recall that there are BCNT arrays of ACNT elements]. DBIDX is always used regardless of whether DAM is Increment or FIFO mode.
15-0	SBIDX	R	0h	Src B-Idx for Dest FIFO Register Set. Value copied from PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array [recall that there are BCNT arrays of ACNT elements]. SBIDX is always used regardless of whether SAM is Increment or FIFO mode.

#### 11.8.1.2.44 DFMPPRXY1 Register (Offset = 354h) [reset = X]

DFMPPRXY1 is shown in [Figure 11-190](#) and described in [Table 11-196](#).

Return to the [Table 11-152](#).

Dst FIFO Set Mem Protect Proxy

**Figure 11-190. DFMPPRXY1 Register**

31	30	29	28	27	26	25	24
RESERVED							
R-X							
23	22	21	20	19	18	17	16
RESERVED							
R-X							
15	14	13	12	11	10	9	8
RESERVED						SECURE	PRIV
R-X						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED				PRIVID			
R-X				R-0h			

**Table 11-196. DFMPPRXY1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R	X	
9	SECURE	R	0h	Secure Level: Deprecated, always read as 0.
8	PRIV	R	0h	Privilege Level: PRIV = 0 : User level privilege PRIV = 1 : Supervisor level privilege PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register [trigger register]. The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the PRIV of the external host that sets up the DMA transaction.
7-4	RESERVED	R	X	
3-0	PRIVID	R	0h	Privilege ID: PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register [trigger register]. The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform memory protection checks based on the privid of the external host that sets up the DMA transaction.



### 11.8.1.2.45 DFBCNT1 Register (Offset = 358h) [reset = X]

DFBCNT1 is shown in [Figure 11-191](#) and described in [Table 11-197](#).

Return to the [Table 11-152](#).

Dst FIFO Set B-Count

**Figure 11-191. DFBCNT1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCNT															
R-X																R-0h															

**Table 11-197. DFBCNT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	X	
15-0	BCNT	R	0h	<p>B-Count Remaining for Dst Register Set:</p> <p>Number of arrays to be transferred where each array is ACNT in length.</p> <p>Represents the amount of data remaining to be written.</p> <p>Initial value is copied from PCNT.</p> <p>TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.</p>

Chapter 12

# Multi-Buffered Serial Peripheral Interface Module (MibSPI) with Parallel Pin Option (MibSPIP)



This chapter provides the specifications for a 16-bit configurable synchronous multi-buffered multi-pin serial peripheral interface (MibSPI). This chapter also provides the specifications for MibSPI with Parallel Pin Option (MibSPIP). The MibSPI is a programmable-length shift register used for high-speed communication between external peripherals or other microcontrollers.

Throughout this chapter, all references to SPI also apply to MibSPI/MibSPIP, unless otherwise noted.

**Note**

This chapter describes a superset implementation of the MibSPI/SPI modules that includes features and functionality that may not be available on some devices. Device-specific content that should be determined by referencing the datasheet includes DMA functionality, MibSPI RAM size, number of transfer groups, number of chip selects, parallel mode support, and availability of 5-pin operation (SPIENA).

<b>12.1 Overview</b>	<b>3709</b>
<b>12.2 Operating Modes</b>	<b>3710</b>
<b>12.3 Test Features</b>	<b>3731</b>
<b>12.4 General-Purpose I/O</b>	<b>3733</b>
<b>12.5 Interrupts</b>	<b>3734</b>
<b>12.6 DMA Interface</b>	<b>3737</b>
<b>12.7 Module Configuration</b>	<b>3738</b>
<b>12.8 Control Registers</b>	<b>3740</b>
<b>12.9 Multi-Buffer RAM</b>	<b>3818</b>
<b>12.10 Parity Memory</b>	<b>3826</b>
<b>12.11 MibSPI Pin Timing Parameters</b>	<b>3829</b>

## 12.1 Overview

The MibSPI/MibSPIP is a high-speed synchronous serial input/output port that allows a serial bit stream of programmed length (2 to 16 bits) to be shifted in and out of the device at a programmed bit-transfer rate. Typical applications for the SPI include interfacing to external peripherals, such as I/Os, memories, display drivers, and analog-to-digital converters.

The SPI has the following attributes:

- 16-bit shift register
- Receive buffer register
- 11-bit baud clock generator
- Serial clock (SPICLK) pin
- 1 SPIPOCI/SPIPICO pin for data transfer, with programmable pin direction
- SPI enable (  $\overline{\text{SPIENA}}$  ) pin
- Up to 6 peripheral chip select (  $\overline{\text{SPICS}}$  ) pins
- SPICLK can be internally-generated (and driven) or received from an external clock source
- Each word transferred can have a unique format
- SPI pins can be used as functional or digital Input/Output pins (GIOs)

---

### Note

PICO - Peripheral In Controller Out Pin  
POCI - Peripheral Out Controller In Pin  
SPICS - SPI Chip Select Pin  
SPIENA - SPI Enable Pin

---

### 12.1.1 Word Format Options

Each word transferred can have a unique format. Several format characteristics are programmable for each word transferred:

- SPICLK frequency
- Character length (2 to 16 bits)
- Phase (with and without delay)
- Polarity (high or low)
- Parity enabled/disabled
- Chip Select(CS) timers for setup and hold
- Shift direction (Most-Significant Bit (MSB) first or Least-Significant Bit (LSB) first)
- Multi-pin parallel modes

### 12.1.2 Multi-buffering (Mib) Support

The MibSPI has a programmable buffer memory that enables programmed transmission to be completed without CPU intervention. The buffers are combined in different Transfer Groups (TGs) that can be triggered by external events (timers, Input/Output activity, and so on) or by the internal tick counter. The internal tick counter supports periodic trigger events. Each buffer of the MibSPI can be associated with different DMA channels in different TGs, allowing the user to move data between internal memory and an external controller with minimal CPU interaction.

### 12.1.2.1 Multi-buffer Mode

Multi-buffer Mode is an extension to the SPI. In multi-buffer mode, many extended features are configurable:

- Number of buffers for each peripheral (or data source/destination, up to 128 buffers supported) or group (up to 8 groupings)
- Triggers for each group, trigger types, trigger sources for individual groups (14 external trigger sources and 1 internal trigger source supported)
- Memory fault detection via an internal parity circuit
- Number of DMA-controlled buffers and number of DMA request channels (up to 8 for each of transmit and receive)
- Number of DMA transfers for each buffer (up to 65536 words for up to 8 buffers)
- Uninterrupted DMA buffer transfer (NOBREAK buffer)

### 12.1.2.2 Compatibility Mode

Compatibility Mode of the MibSPI makes it behave exactly like a standard platform SPI module and ensures full compatibility with other SPIs. All features in compatibility mode of the MibSPI are directly applicable to a SPI. Multi-buffer Mode features are not available in Compatibility Mode.

---

#### Note

The SPIDAT0 register is not accessible in the multi-buffer mode of MibSPI. It is only accessible in compatibility mode.

---

### 12.1.3 Transmission Lock (Multi-Buffer Mode Controller Only)

Some peripheral devices require transmission of a command followed by data. In this case the SPI transaction should not be interrupted by another group transfer. The LOCK bit within each buffer allows a consecutive transfer to happen without being interrupted by another higher-priority group transfer.

## 12.2 Operating Modes

The SPI can be configured via software to operate as either a controller or a peripheral. The CONTROLLER bit (SPIGCR1[0]) selects the configuration of the SPIPICO and SPIPOCI pins. CLKMOD bit (SPIGCR1[1]) determines whether an internal or external clock source will be used.

The chip select ( $\overline{\text{SPICS}}$ ) pins are used when communicating with multiple peripheral devices or, with a single peripheral device, to delimit messages containing a leading register address. When a write occurs to SPIDAT1 in controller mode, the  $\overline{\text{SPICS}}$  pins are automatically driven to select the specified peripheral device.

Handshaking mechanism, provided by the  $\overline{\text{SPIENA}}$  pin, enables a peripheral SPI to delay the generation of the clock signal supplied by the controller if it is not prepared for the next exchange of data.

---

#### Note

If in the peripheral mode of operation and configured in either 3-pin or 4-pin (without  $\overline{\text{SPIENA}}$ ) modes, there must be a minimum of 8 VCLK cycles of delay between the last SPICLK and the start of the SPICLK for the next buffer transmit. In general, this equates to a VCLK/SPICLK ratio of  $\leq 16$  requiring a minimum of 1 SPICLK delay between transmissions.

---

### 12.2.1 Pin Configurations

The SPI supports data connections as shown in [Table 12-1](#).

#### Note

1. When the SPICS signals are disabled, the chip select field in the transmit data is not used.
2. When the SPIENA signal is disabled, the SPIENA pin is ignored in controller mode, and not driven as part of the SPI transaction in peripheral mode.

**Table 12-1. Pin Configurations**

Pin	Controller Mode		Peripheral Mode	
SPICLK	Drives the clock to external devices		Receives the clock from the external controller	
SPISOC	Receives data from the external peripheral		Sends data to the external controller	
SPICPO	Transmits data to the external peripheral		Receives data from the external controller	
SPIENA	<b>SPIENA disabled:</b> GIO	<b>SPIENA enabled:</b> Receives ENA signal from the external peripheral	<b>SPIENA disabled:</b> GIO	<b>SPIENA enabled:</b> Drives ENA signal from the external controller
SPICS	<b>SPICS disabled:</b> GIO	<b>SPICS enabled:</b> Selects one or more peripheral devices	<b>SPICS disabled:</b> GIO	<b>SPICS enabled:</b> Receives the CS signal from the external controller

### 12.2.2 Data Handling

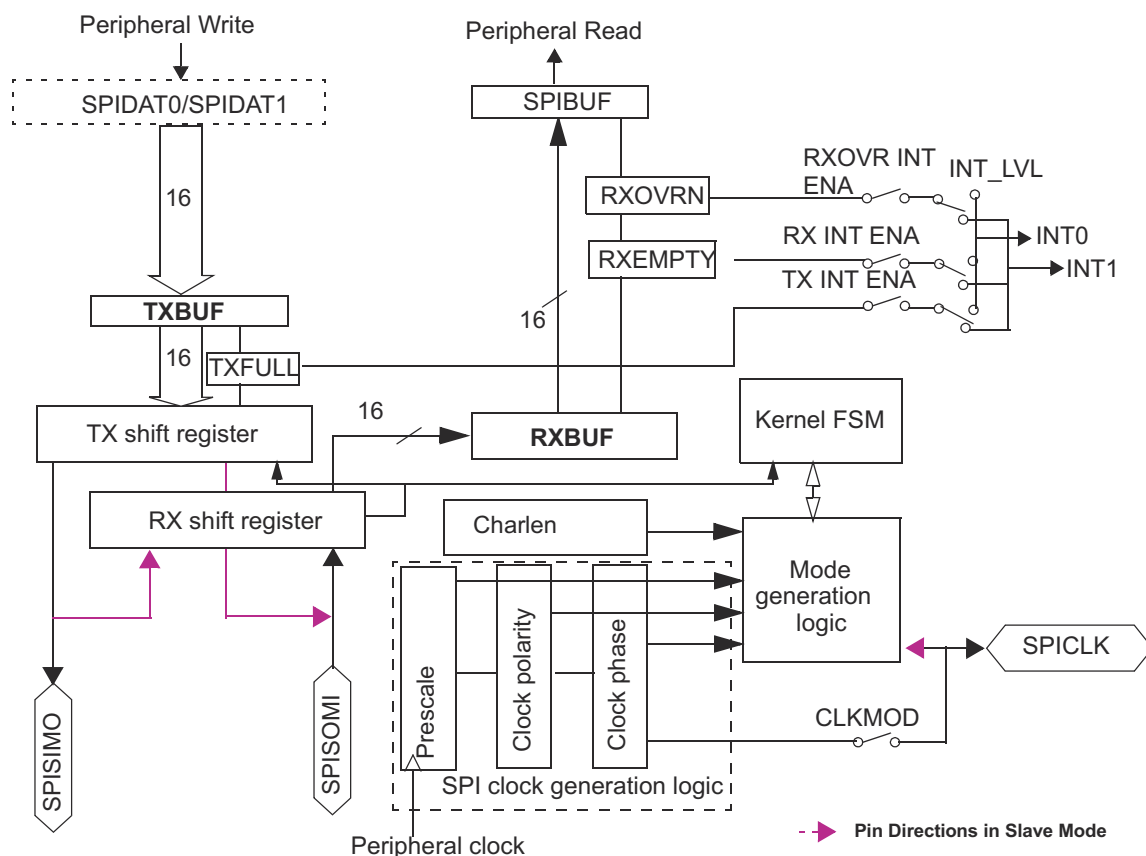
[Figure 12-1](#) shows the SPI transaction hardware. TXBUF and RXBUF are internal buffers that are intended to improve the overall throughput of data transfer. TXBUF is a transmit buffer, while RXBUF is a receive buffer.

#### 12.2.2.1 Data Sequencing when SPIDAT0 or SPIDAT1 is Written

- If both the TX shift register and TXBUF are empty, then the data is directly copied to the TX shift register. For devices with DMA, if DMA is enabled, a transmit DMA request (TX\_DMA\_REQ) is generated to cause the next word to be fetched. If transmit interrupts are enabled, a transmitter-empty interrupt is generated.
- If the TX shift register is already full or is in the process of shifting and if TXBUF is empty then the data written to SPIDAT0 / SPIDAT1 is copied to TXBUF and TXFULL flag is set to 1 at the same time.
- When a shift operation is complete, data from the TXBUF (if it is full) is copied into TX shift register and the TXFULL flag is cleared to 0 to indicate that next data can be fetched. A transmit DMA request (if enabled) or a transmitter-empty interrupt (if enabled) is generated at the same time.

#### 12.2.2.2 Data Sequencing when All Bits Shifted into RXSHIFT Register

- If both SPIBUF and RXBUF are empty, the received data in RX shift register is directly copied into SPIBUF and the receive DMA request (if enabled) is generated and the receive-interrupt (if enabled) is generated. The RXEMPTY flag in SPIBUF is cleared at the same time.
- If SPIBUF is already full at the end of receive completion, the RX shift register contents is copied to RXBUF. A receive DMA request is generated, if enabled. The receive complete interrupt line remains high.
- If SPIBUF is read by the CPU or DMA and if RXBUF is full, then the contents of RXBUF are copied to SPIBUF as soon as SPIBUF is read. RXEMPTY flag remains cleared, indicating that SPIBUF is still full.
- If both SPIBUF and RXBUF are full, then RXBUF will be overwritten and the RXOVR interrupt flag is set and an interrupt is generated, if enabled.



- 1 This is a representative diagram, which shows three-pin mode hardware.
- 2 TXBUF, RXBUF, and SHIFT\_REGISTER are user-invisible registers.
- 3 SPIDAT0 and SPIDAT1 are user-visible, and are physically mapped to the contents of TXBUF.
- 4 SPISIMO, SPISOMI, SPICLK pin directions depend on the Master or Slave Mode.

**Figure 12-1. SPI Functional Logic Diagram**

### 12.2.2.3 Three-Pin Mode

In controller mode configuration (MASTER = 1 and CLKMOD = 1), the SPI provides the serial clock on the SPICLK pin. Data is transmitted on the SPISIMO pin and received on the SPISOMI pin (see Figure 12-2).

Data written to the shift register (SPIDAT0 / SPIDAT1) initiates data transmission on the SPISIMO pin, MSB first. Simultaneously, received data is shifted through the SPISOMI pin into the LSB of the SPIDAT0 register. When the selected number of bits have been transmitted, the received data in the shift register is transferred to the SPIBUF register for the CPU to read. Data is stored right-justified in SPIBUF.

See Section 12.2.2.1 and Section 12.2.2.2 for details about the data handling for transmit and receive operations.

In peripheral mode configuration (MASTER = 0 and CLKMOD = 0), data shifts out on the SPIPOCI pin and in on the SPIPICO pin. The SPICLK pin is used as the input for the serial shift clock, which is supplied from the external network controller. The transfer rate is defined by this clock.

Data written to the SPIDAT0 or SPIDAT1 register is transmitted to the network when the SPICLK signal is received from the network controller. To receive data, the SPI waits for the network controller to send the SPICLK signal and then shifts data on the SPISIMO pin into the RX shift register. If data is to be transmitted by the peripheral simultaneously, it must be written to the SPIDAT0 or SPIDAT1 register before the beginning of the SPICLK signal.

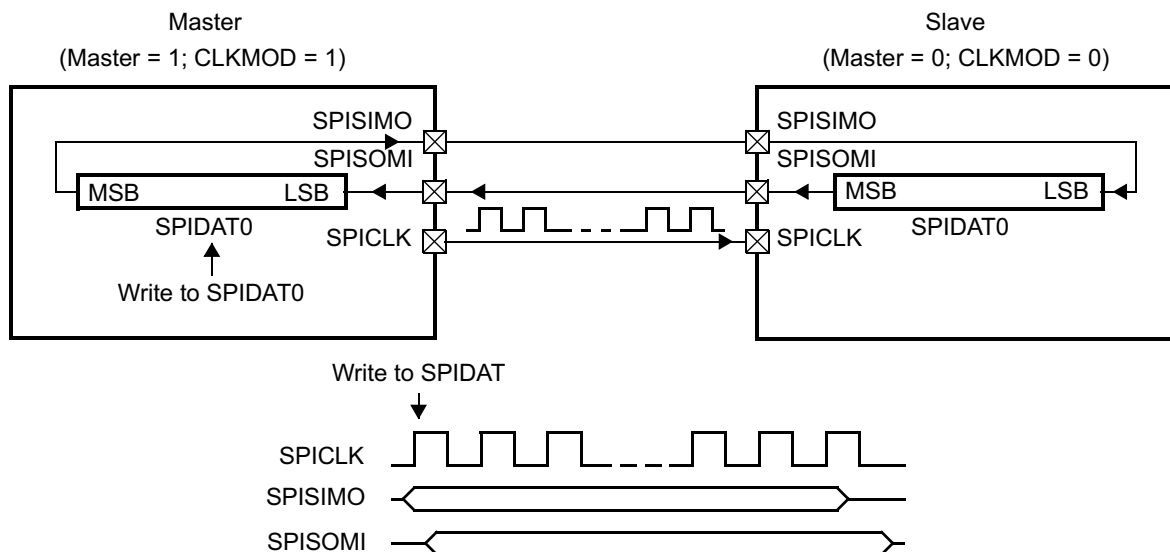


Figure 12-2. SPI Three-Pin Operation

## 12.2.3 Operation with $\overline{\text{SPICS}}$

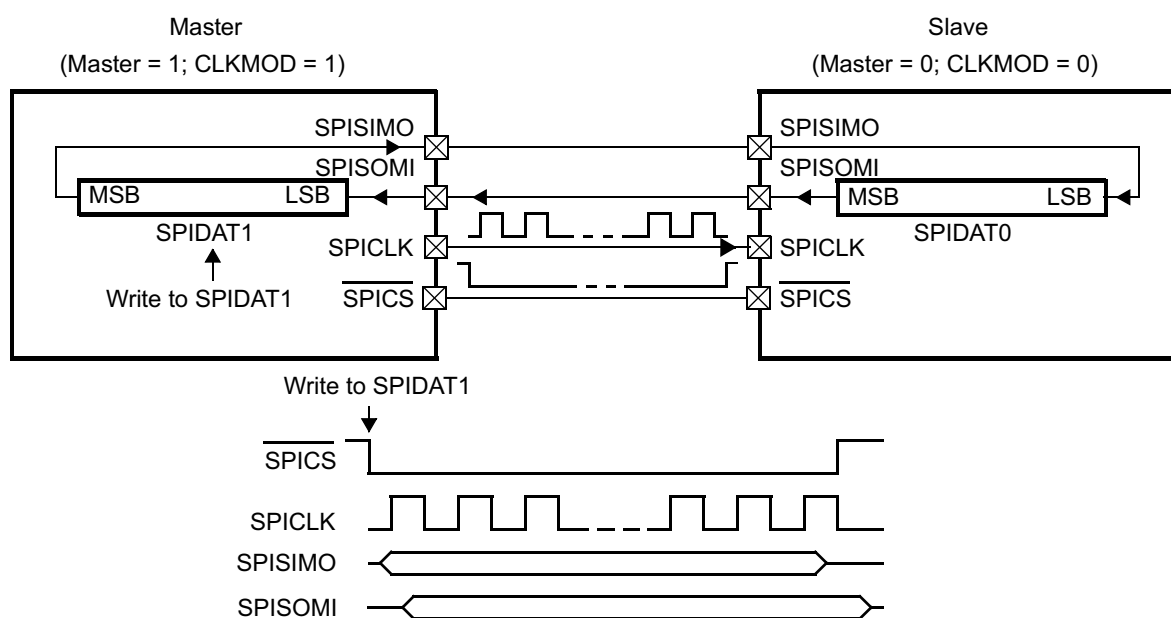
In controller mode, each chip select signal is used to select a specific peripheral. In peripheral mode, the chip select signal is used to enable and disable the transfer. Chip-select functionality is enabled by setting one of the  $\overline{\text{SPICS}}$  pins as a chip select. It is disabled by setting all  $\overline{\text{SPICS}}$  pins as GPIOs in SPIPC0.

### 12.2.3.1 Multiple Chip Selects

The  $\overline{\text{SPICS}}$  pins that are used must be configured as functional pins in the SPIPC0 register. The default pattern to be put on the  $\overline{\text{SPICS}}$  when all the peripherals are deactivated is set in the SPIDEF register. This pattern allows different peripherals with different chip-select polarity to be activated by the SPI.

The controller-mode SPI is capable of driving either 0 or 1 as the active value for any  $\overline{\text{SPICS}}$  output pin. The drive state for the  $\overline{\text{SPICS}}$  pins is controlled by the CSNR field of SPIDAT1. The pattern that is driven will select the peripheral to which the transmission is dedicated.

In peripheral mode, the SPI can only be selected by an active value of 0 on any of its selected  $\overline{\text{SPICS}}$  input pins.



**Figure 12-3. Operation with  $\overline{\text{SPICS}}$**



### 12.2.4 Operation with $\overline{\text{SPIENA}}$

The  $\overline{\text{SPIENA}}$  operates as a WAIT signal pin. For both the peripheral and the controller, the  $\overline{\text{SPIENA}}$  pin must be configured to be functional ( $\text{SPIPC0}[8] = 1$ ). In this mode, an active-low signal from the peripheral on the  $\overline{\text{SPIENA}}$  pin allows the controller SPI to drive the clock pulse stream. A high signal tells the controller to hold the clock signal (and delay SPI activity).

If the  $\overline{\text{SPIENA}}$  pin is in high-impedance mode ( $\text{ENABLE\_HIGHZ} = 1$ ), the peripheral will put  $\overline{\text{SPIENA}}$  into the high-impedance once it completes receiving a new character. If the  $\overline{\text{SPIENA}}$  pin is in push-pull mode ( $\text{ENABLE\_HIGHZ} = 0$ ), the peripheral will drive  $\overline{\text{SPIENA}}$  to 1 once it completes receiving a new character. The peripheral will drive  $\overline{\text{SPIENA}}$  low again for the next word to transfer, after new data is written to the peripheral TX shift register.

In controller mode ( $\text{CLKMOD} = 1$ ), if the  $\overline{\text{SPIENA}}$  pin is configured as functional, then the pin acts as an input pin. If configured as a peripheral SPI and as functional, the  $\overline{\text{SPIENA}}$  pin acts as an output pin.

#### Note

During a transfer, if a peripheral-mode SPI detects a deassertion of its chip select before its internal character length counter overflows, then it places  $\text{SPISOMI}$  and  $\overline{\text{SPIENA}}$  (if  $\text{ENABLE\_HIGHZ}$  bit is set to 1) in high-impedance mode. Once this condition has occurred, if a  $\text{SPICLK}$  edge is detected while the chip select is deasserted, then the SPI stops that transfer and sets an  $\text{DLENERR}$  error flag and generates an interrupt (if enabled).

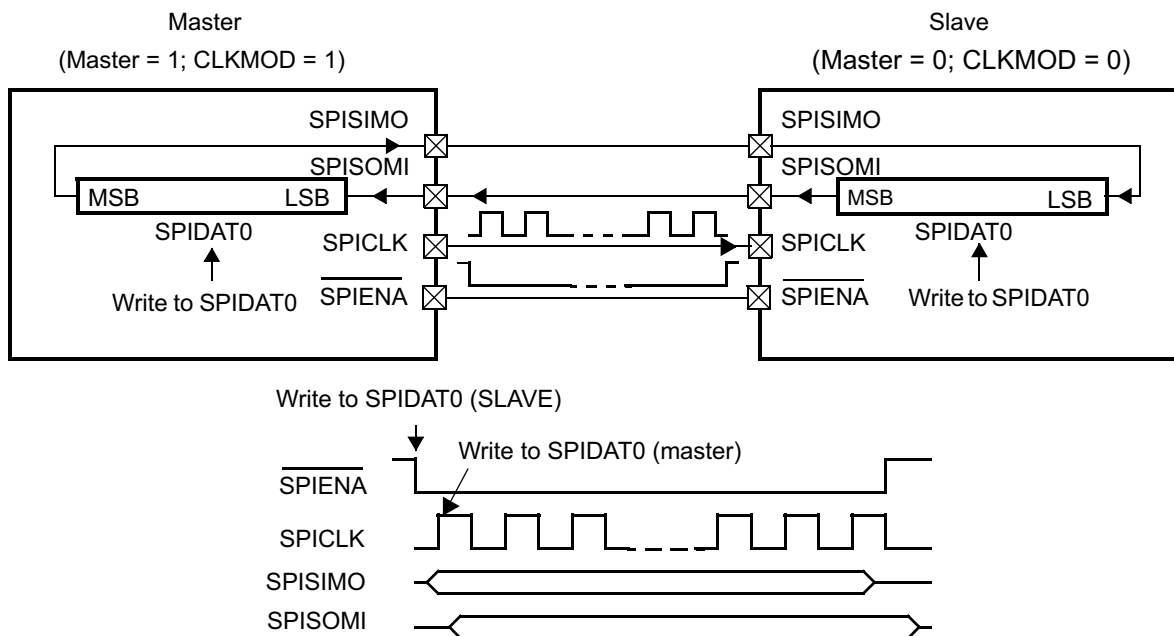


Figure 12-4. Operation with  $\overline{\text{SPIENA}}$

### 12.2.5 Five-Pin Operation (Hardware Handshaking)

Five-pin operation combines the functionality of three-pin mode, plus the enable pin and one or more chip select pins. The result is full hardware handshaking. To use this mode, both the  $\overline{\text{SPIENA}}$  pin and the required number of  $\overline{\text{SPICS}}$  pins must be configured as functional pins.

If the  $\overline{\text{SPIENA}}$  pin is in high-impedance mode ( $\text{ENABLE\_HIGHZ} = 1$ ), the peripheral SPI will put this signal into the high-impedance state by default. The peripheral will drive the signal  $\overline{\text{SPIENA}}$  low when new data is written to the peripheral shift register and the peripheral has been selected by the controller ( $\overline{\text{SPICS}}$  is low).

If the  $\overline{\text{SPIENA}}$  pin is in push-pull mode ( $\text{ENABLE\_HIGHZ} = 0$ ), the peripheral SPI drives this pin high by default when it is in functional mode. The peripheral SPI will drive the  $\overline{\text{SPIENA}}$  signal low when new data is written to the peripheral shift register ( $\text{SPIDAT0}/\text{SPIDAT1}$ ) and the peripheral is selected by the controller ( $\overline{\text{SPICS}}$  is low). If the peripheral is deselected by the controller ( $\overline{\text{SPICS}}$  goes high), the peripheral  $\overline{\text{SPIENA}}$  signal is driven high.

#### Note

Push-pull mode of the  $\overline{\text{SPIENA}}$  pin can be used only when there is a single peripheral in the system. When multiple SPI peripheral devices are connected to the common  $\overline{\text{SPIENA}}$  pin, all of the peripherals should configure their  $\overline{\text{SPIENA}}$  pins in high-impedance mode.

In controller mode, if the  $\overline{\text{SPICS}}$  pins are configured as functional pins, then the pins will be in output mode. A write to the controller's  $\text{SPIDAT1}/\text{SPIDAT0}$  register will automatically drive the  $\overline{\text{SPICS}}$  signals low. The controller will drive the  $\overline{\text{SPICS}}$  signals high again after completing the transfer of the bits of the data.

In peripheral mode ( $\text{CLKMOD} = 0$ ), the  $\overline{\text{SPICS}}$  pins act as SPI functional inputs.

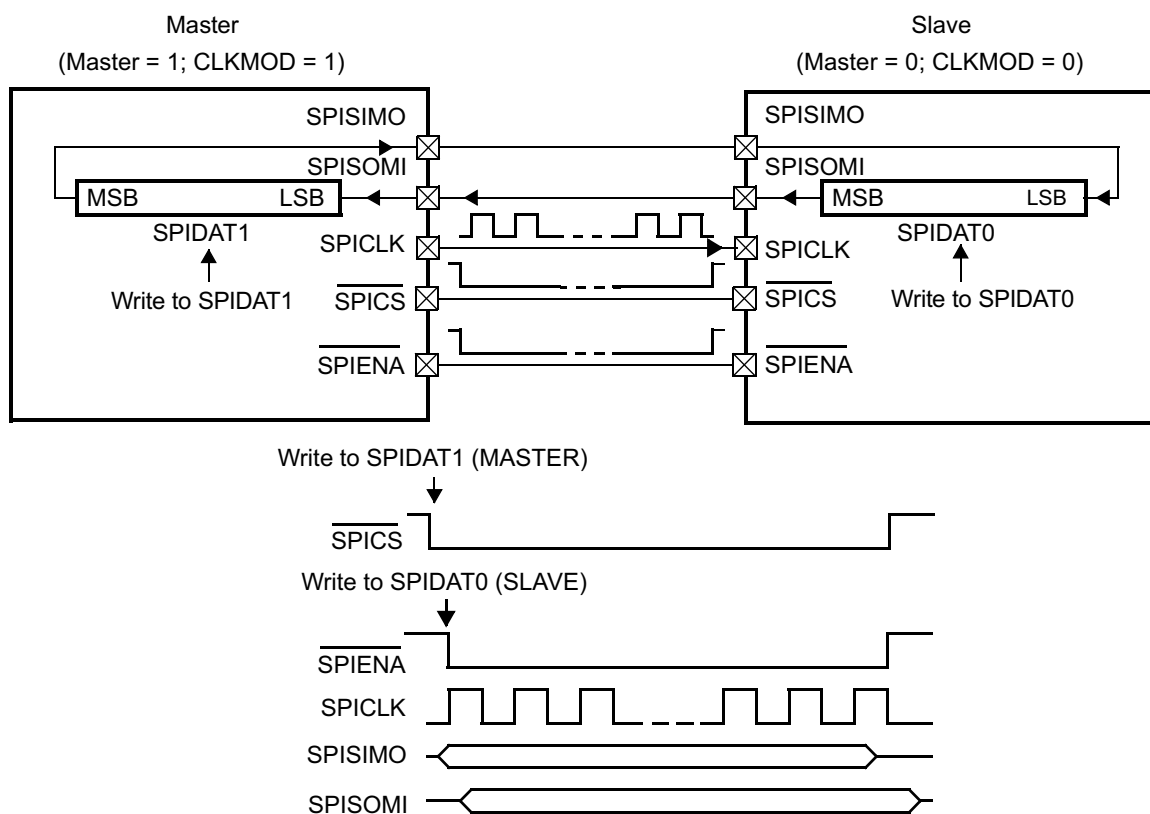


Figure 12-5. SPI Five-Pin Option with  $\overline{\text{SPIENA}}$  and  $\overline{\text{SPICS}}$

### 12.2.6 Data Formats

To support multiple different types of peripherals in one SPI network, four independent data word formats are implemented that allow configuration of individual data word length, polarity, phase, and bit rate. Each word transmitted can select which data format to use via the bits DFSEL[1:0] in its control field from one of the four data word formats. Same data format can be supported on multiple chip selects.

Data formats 0, 1, 2, and 3 can be configured through SPIFMTx control registers.

Each SPI data format includes the standard SPI data format with enhanced features:

- Individually-configurable shift direction can be used to select MSB first or LSB first, whereas the position of the MSB depends on the configured data word length.
- Receive data is automatically right-aligned, independent of shift direction and data word length. Transmit data has to be written right-aligned into the SPI and the internal shift register will transmit according to the selected shift direction and data word length for correct transfer.
- To increase fault detection of data transmission and reception, an odd or even parity bit can be added at the end of a data word. The parity generator can be enabled or disabled individually for each data format. If a received parity bit does not match with the locally calculated parity bit, the parity error flag (PARITYERR) is set and an interrupt is asserted (if enabled).

Since the controller-mode SPI can drive two consecutive accesses to the same peripheral, an 8-bit delay counter is available to satisfy the delay time for data to be refreshed in the accessed peripheral. The delay counter can be programmed as part of the data format.

CHARLEN[4:0] specifies the number of bits (2 to 16) in the data word. The CHARLEN[4:0] value directs the state control logic to count the number of bits received or transmitted to determine when a complete word is transferred.

Data word length **must** be programmed to the same length for both the **controller** and the **peripheral**. However, when chip selects are used, there may be multiple targets with different lengths in the system.

#### Note

Data must be right-justified when it is written to the SPI for transmission irrespective of its character length or word length.

Figure 12-6 shows how a 12-bit word (0xEC9) needs to be written to the transmit buffer to be transmitted correctly.

**Figure 12-6. Format for Transmitting an 12-Bit Word**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
x	x	x	x	1	1	1	0	1	1	0	0	1	0	0	1

#### Note

The received data is always stored right-justified regardless of the character length or direction of shifting and is padded with leading 0s when the character length is less than 16 bits.

Figure 12-7 shows how a 10-bit word (0x0A2) is stored in the buffer once it is received.

**Figure 12-7. Format for Receiving an 10-Bit Word**

D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0
0	0	0	0	0	0	0	0	1	0	1	0	0	0	1	0

## 12.2.7 Clocking Modes

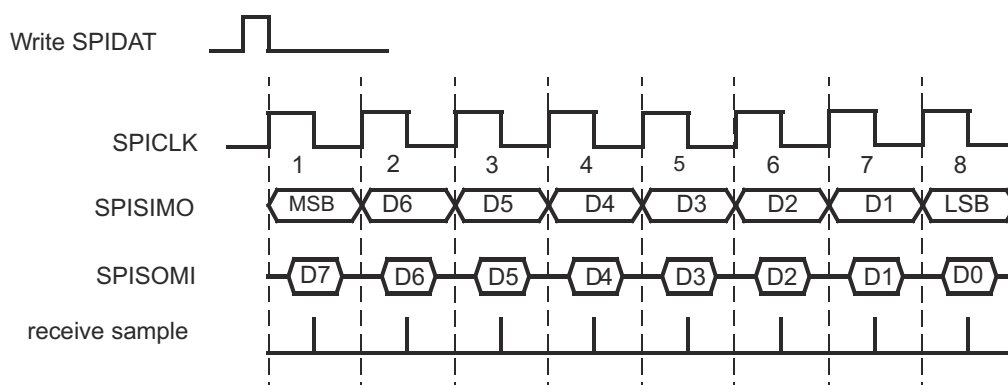
**SPICLK** may operate in four different modes, depending on the choice of phase (delay/no delay) and the polarity (rising edge/falling edge) of the clock.

The data input and output edges depend on the values of both **POLARITY** and **PHASE** as shown in Table 12-2.

**Table 12-2. Clocking Modes**

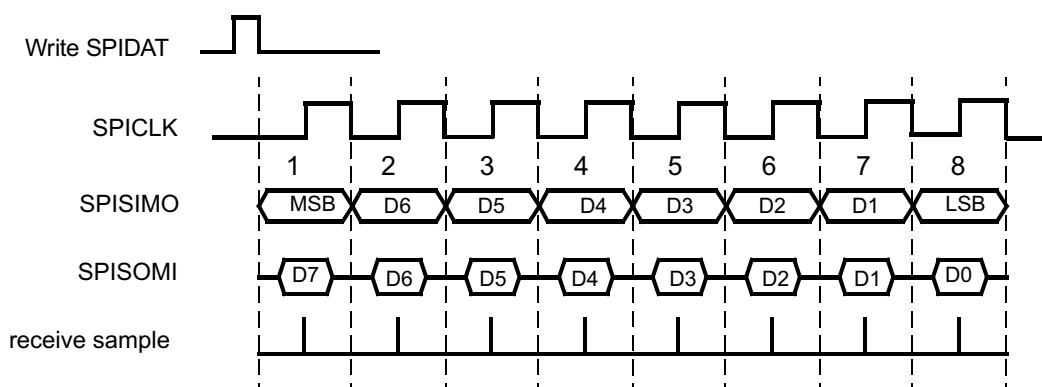
POLARITY	PHASE	Action
0	0	Data is output on the rising edge of SPICLK. Input data is latched on the falling edge.
0	1	Data is output one half-cycle before the first rising edge of SPICLK and on subsequent falling edges. Input data is latched on the rising edge of SPICLK.
1	0	Data is output on the falling edge of SPICLK. Input data is latched on the rising edge.
1	1	Data is output one half-cycle before the first falling edge of SPICLK and on subsequent rising edges. Input data is latched on the falling edge of SPICLK.

Figure 12-8 to Figure 12-11 illustrate the four possible configurations of **SPICLK** corresponding to each mode. Having four signal options allows the SPI to interface with many different types of serial devices.



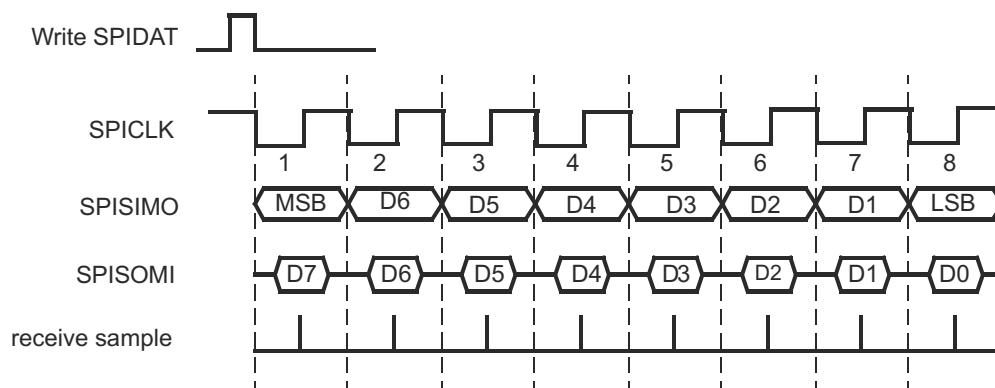
Data is output on the rising edge of SPICLK.  
Input data is latched on the falling edge of SPICLK.

**Figure 12-8. Clock Mode with Polarity = 0 and Phase = 0**



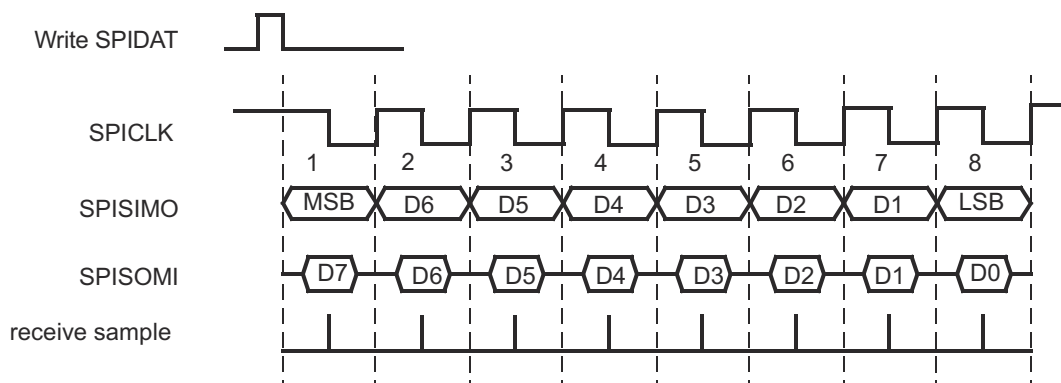
Data is output one-half cycle before the first rising edge of SPICLK and on subsequent falling edges of SPICLK  
Input data is latched on the rising edge of SPICLK

**Figure 12-9. Clock Mode with Polarity = 0 and Phase = 1**



Data is output on the falling edge of SPICLK.  
Input data is latched on the rising edge of SPICLK.

**Figure 12-10. Clock Mode with Polarity = 1 and Phase = 0**



Data is output one-half cycle before the first falling edge of SPICLK and on the subsequent rising edges of SPICLK.  
Input data is latched on the falling edge of SPICLK.

**Figure 12-11. Clock Mode with Polarity = 1 and Phase = 1**

### 12.2.8 Data Transfer Example

Figure 12-12 illustrates a SPI data transfer between two devices using a character length of five bits.

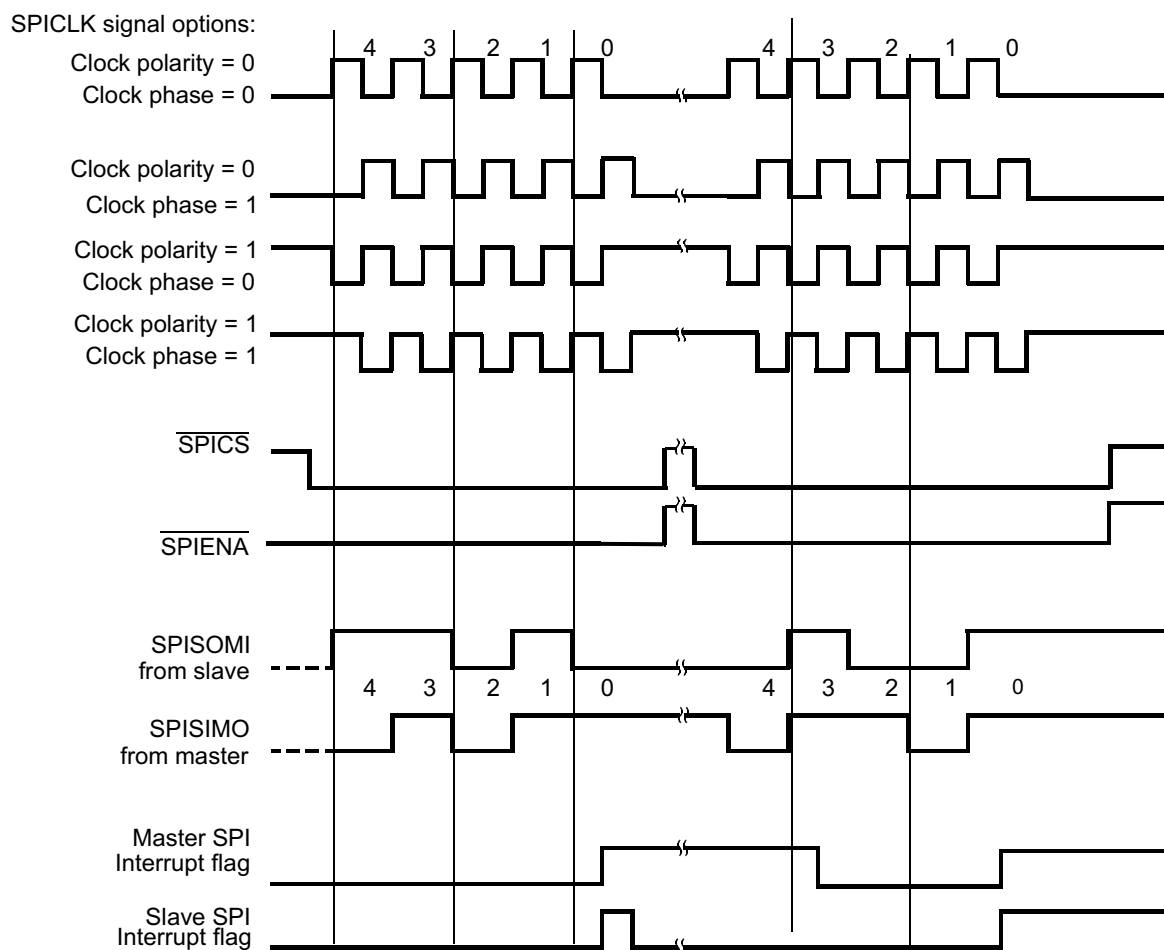


Figure 12-12. Five Bits per Character (5-Pin Option)

### 12.2.9 Decoded and Encoded Chip Select (Controller Only)

In this device, the SPI can connect to up to 6 individual peripheral devices using chip-selects by routing one wire to each peripheral. The 6 chip selects in the control field are directly connected to the 6 pins. The default value of each chip select (not active) can be configured via the register CSDEF. During a transmission, the value of the chip select control field (CSNR) of the SPIDAT1 register (SPIDAT1) is driven on the  $\overline{\text{SPICS}}$  pins. When the transmission finishes, the default chip-select value (defined by the CSDEF register) is put on the  $\overline{\text{SPICS}}$  pins.

The SPI can support more than 6 slaves by using encoded chip selects. To connect the SPI with encoded slaves devices, the CSNR field allows multiple active  $\overline{\text{SPICS}}$  pins at the same time, which enables encoded chip selects from 0 to 16. To use encoded chip selects, all 6 chip select lines have to be connected to each peripheral device and each peripheral needs to have a unique chip-select address. The CSDEF register is used to provide the address at which slaves devices are all de-selected.

Users can combine decoded and encoded chip selects. For example,  $n$   $\overline{\text{SPICS}}$  pins can be used for encoding an  $n$ -bit address and the remaining pins can be connected to decoded-mode slaves.

### 12.2.10 Variable Chip Select Setup and Hold Timing (Controller Only)

In order to support slow peripheral devices, a delay counter can be configured to delay data transmission after the chip select is activated. A second delay counter can be configured to delay the chip select deactivation after the last data bit is transferred. Both delay counters are clocked with the peripheral clock (VCLK).

If a particular data format specifically does not require these additional set-up or hold times for the chip select pins, then they can be disabled in the corresponding SPIFMTx register.

### 12.2.11 Hold Chip-Select Active

Some peripheral devices require the chip select signal to be held continuously active during several consecutive data word transfers. Other peripheral devices require the chip select signal to be deactivated between consecutive data word transfers.

CSHOLD is programmable in both controller and peripheral modes of the multi-buffer mode of SPI. However, the meaning of CSHOLD in controller mode and peripheral mode are different.

#### Note

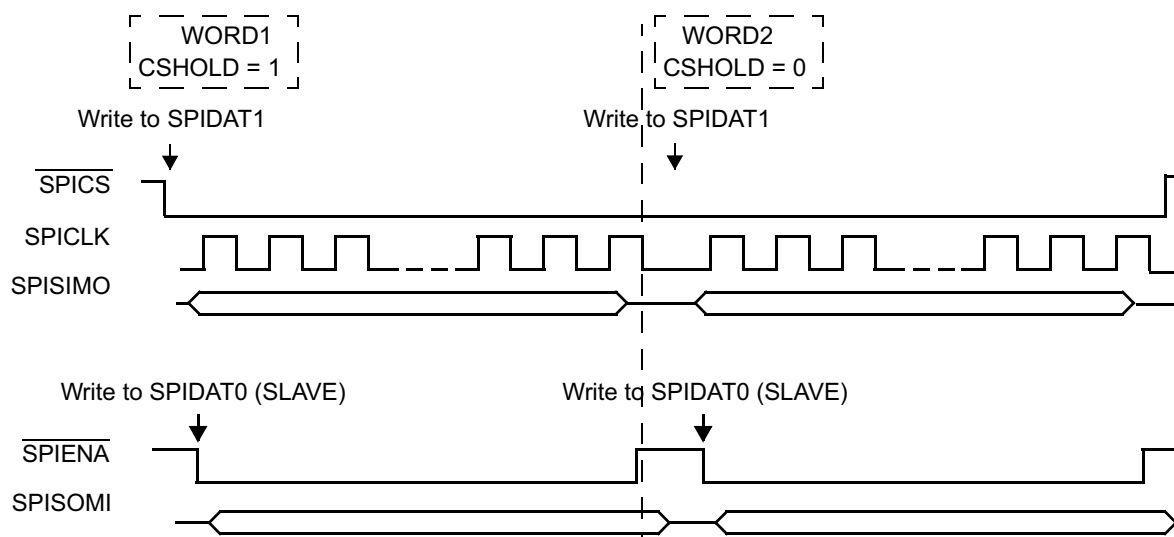
If the CSHOLD bit is set within the current data control field, the programmed hold time and the following programmed set-up time will not be applied between transactions.

#### 12.2.11.1 CSHOLD Bit in Controller Mode

Each word in a controller-mode SPI can be individually initialized for one of the two modes via the CSHOLD bit in its control field.

If the CSHOLD bit is set in the control field of a word, the chip select signal will not be deactivated until the next control field is loaded with new chip select information. Since the chip-select is maintained active between two transfers, the chip-select hold delay (T2CDELAY) is not applied at the end of the current transaction, and the chip-select set-up time delay (C2TDELAY) is not applied as well at the beginning of the following transaction. However, the wait delay (WDELAY) will be still applied between the two transactions, if the WDEL bit is set within the control field.

Figure 12-13 shows the SPI pins when a controller-mode SPI transfers a word that has its CSHOLD bit set. The chip-select pins will not be deasserted after the completion of this word. If the next word to transmit has the same chip-select number (CSNR) value, the chip select pins will be maintained until the completion of the second word, regardless of whether the CSHOLD bit is set or not.



**Figure 12-13. Typical Diagram when a Buffer in controller is in CSHOLD Mode (SPI-SPI)**

### 12.2.11.2 CSHOLD Bit in Peripheral Mode (Multi-buffered Mode)

If the CSHOLD bit in a buffer is set to 1, then the MibSPI does not wait for the  $\overline{\text{SPICS}}$  pins to be de-asserted at the end of the shift operation to copy the received data to the receive RAM. With this feature, it is possible for a peripheral in multi-buffer mode to do multiple data transfers without requiring the  $\overline{\text{SPICS}}$  pins to be deasserted between two buffer transfers.

If the CSHOLD bit in a buffer is cleared to 0 in a peripheral MibSPI, even after the shift operation is done, the MibSPI waits until the  $\overline{\text{SPICS}}$  pin (if functional) is deasserted to copy the received data to the RXRAM.

If the CSHOLD bit is maintained as 0 across all the buffers, then the peripheral in multi-buffer mode requires its  $\overline{\text{SPICS}}$  pins to be deasserted between any two buffer transfers; otherwise, the peripheral SPI will be unable to respond to the next data transfer.

#### Note

In compatibility mode, the peripheral does not require the  $\overline{\text{SPICS}}$  pin to be deasserted between two buffer transfers. The CSHOLD bit of the peripheral will be ignored in compatibility mode.

### 12.2.12 Detection of Peripheral Desynchronization (Controller Only)

When a peripheral supports generation of an enable signal (ENA), desynchronization can be detected. With the enable signal a peripheral indicates to the controller that it is ready to exchange data. A desynchronization can occur if one or more clock edges are missed by the peripheral. In this case, the peripheral may block the SOMI line until it detects clock edges corresponding to the next data word. This would corrupt the data word of the desynchronized peripheral and the consecutive data word. A configurable 8-bit time-out counter (T2EDELAY), which is clocked with SPICLK, is implemented to detect this peripheral malfunction. After the transmission has finished (end of last bit transferred: either last data bit or parity bit) the counter is started. If the ENA signal generated by the peripheral does not become inactive before the counter overflows, the DESYNC flag is set and an interrupt is asserted (if enabled).



---

### Note

#### Inconsistency of Desynchronization Flag in Compatibility Mode MibSPI

Because of the nature of this error, under some circumstances it is possible for a desync error detected for the previous buffer to be visible in the current buffer. This is due to the fact that receive completion flag/interrupt will be generated when the buffer transfer is completed. But desync will be detected after the buffer transfer is completed. So, if VBUS controller reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct desync condition. This inconsistency in the desync flag is valid only in compatibility mode of MibSPI. In multi-buffer mode, the desync flag is always assured to be for the current buffer.

---

#### 12.2.13 ENA Signal Time-Out (Controller Only)

The SPI in controller mode waits for the hardware handshake signal (ENA) coming from the addressed peripheral before performing a data transfer. To avoid stalling the SPI by a non-responsive peripheral device, a time-out value can be configured using C2EDELAY. If the time-out counter overflows before an active ENA signal is sampled, the TIMEOUT flag in the status register SPIFLG is set and the TIMEOUT flag in the status field of the corresponding buffer is set.

---

### Note

When the chip select signal becomes active, no breaks in transmission are allowed. The next arbitration is performed while waiting for the time-out to occur.

---

#### 12.2.14 Data-Length Error

A SPI can generate an error flag by detecting any mismatch in length of received or transmitted data and the programmed character length under certain conditions.

**Data-Length Error in Controller Mode:** During a data transfer, if the SPI detects a de-assertion of the  $\overline{\text{SPIENA}}$  pin (by the peripheral) while the character counter is not overflowed, then an error flag is set to indicate a data-length error. This can be caused by a peripheral receiving extra clocks (for example, due to noise on the SPICLK line).

---

### Note

In a controller mode SPI, the data length error will be generated only if the  $\overline{\text{SPIENA}}$  pin is enabled as a functional pin.

---

**Data-Length Error in peripheral Mode:** During a transfer, if the SPI detects a de-assertion of the  $\overline{\text{SPICS}}$  pin before its character length counter overflows, then an error flag is set to indicate a data-length error. This situation can arise if the peripheral SPI misses one or more SPICLK pulses from the controller. This error in peripheral mode implies that both the transmitted and received data were not complete.

---

### Note

In a peripheral mode SPI, the data-length error flag will be generated only if at least one of the  $\overline{\text{SPICS}}$  pins are configured as functional, and are being used for selecting the peripheral.

---

#### 12.2.15 Parallel Mode (Multiple PICO/POCI Support, not available on all devices)

In order to increase throughput, the parallel mode of the SPI enables the module to send data over more than one data line (parallel 2, 4, or 8). When parallel mode is used, the data length must be set as 16 bits. Only module MIBSPIP5 supports Parallel Mode.

This feature increases throughput by 2 for 2 pins, by 4 for 4 pins, or by 8 for 8 pins.

Parallel mode supports the following features:

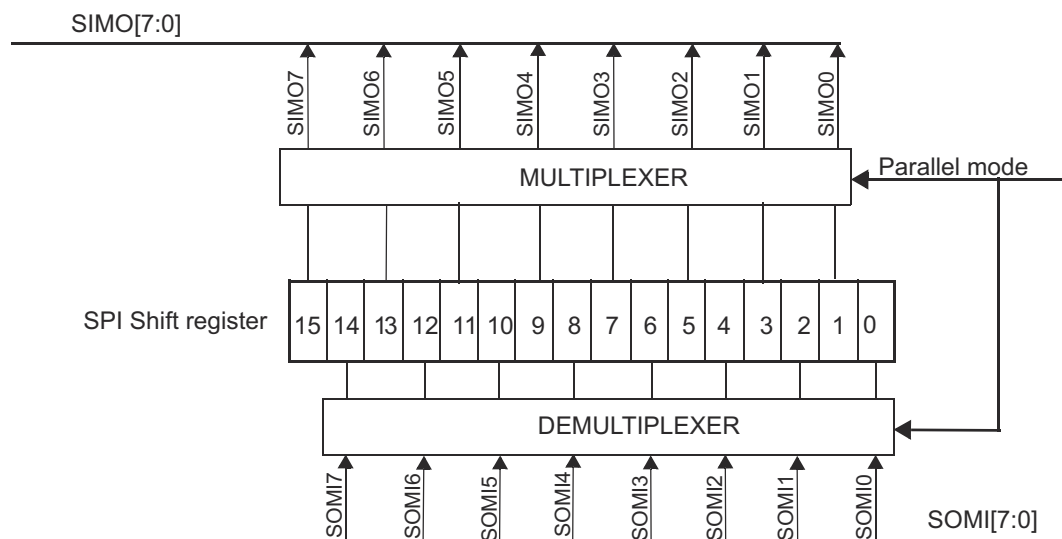
- Scalable data lines (1, 2, 4, 8) per direction. (POCI and PICO lines)
- All clock schemes are supported (clock phase and polarity)
- Parity is supported. The parity bit will be transmitted on bit0 of the PICO/POCI lines. The receive parity is expected on bit0 of the POCI/PICO pins.

Parallel mode can be programmed using the PMODEx bits of SPIPMCTRL register. See [Section 12.8.26](#) for details about this register.

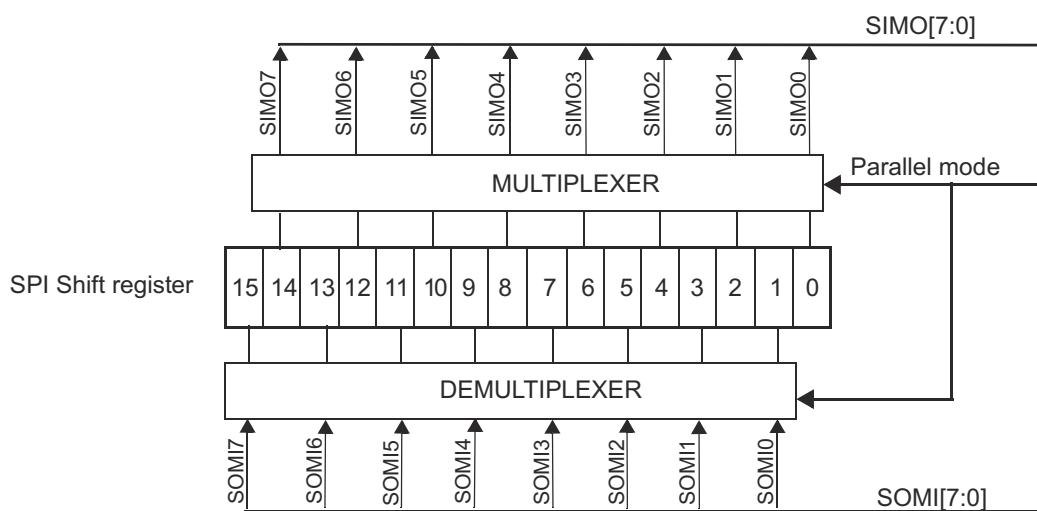
After reset the parallel mode selection bits are cleared (single PICO/POCI lines).

### 12.2.15.1 Parallel Mode Block Diagram

[Figure 12-14](#) and [Figure 12-15](#) show the parallel connections to the SPI shift register.



**Figure 12-14. Block Diagram Shift Register, MSB First**



**Figure 12-15. Block Diagram Shift Register, LSB First**

### 12.2.15.2 Parallel Mode Pin Mapping, MSB First

Table 12-3 and Table 12-4 describe the POCI and PICO pin mapping when the SPI is used in parallel mode (1, 2, 4, 8) pin mode, MSB first.

#### Note

MSB-first or LSB-first can be configured using the SHIFTDIRx bit of the SPIFMTx registers.

**Table 12-3. Pin Mapping for PICO Pin with MSB First**

Parallel Mode	Shift Register Bit	PICO[7:0]
1	15	0
2	15	1
	7	0
4	15	3
	11	2
	7	1
	3	0
8	15	7
	13	6
	11	5
	9	4
	7	3
	5	2
	3	1
	1	0

**Table 12-4. Pin Mapping for POCI Pin with MSB First**

Parallel Mode	Shift Register Bit	POCI[7:0]
1	0	0
2	0	0
	8	1
4	0	0
	4	1
	8	2
	12	3
8	0	0
	2	1
	4	2
	6	3
	8	4
	10	5
	12	6
	14	7

### 12.2.15.3 Parallel Mode Pin Mapping, MSB-First, LSB-First

Table 12-5 and Table 12-6 describe the PICO and POCI pin mapping when SPI is used in parallel mode (1, 2, 4, 8) pin mode, LSB first.

**Table 12-5. Pin Mapping for PICO Pin with LSB First**

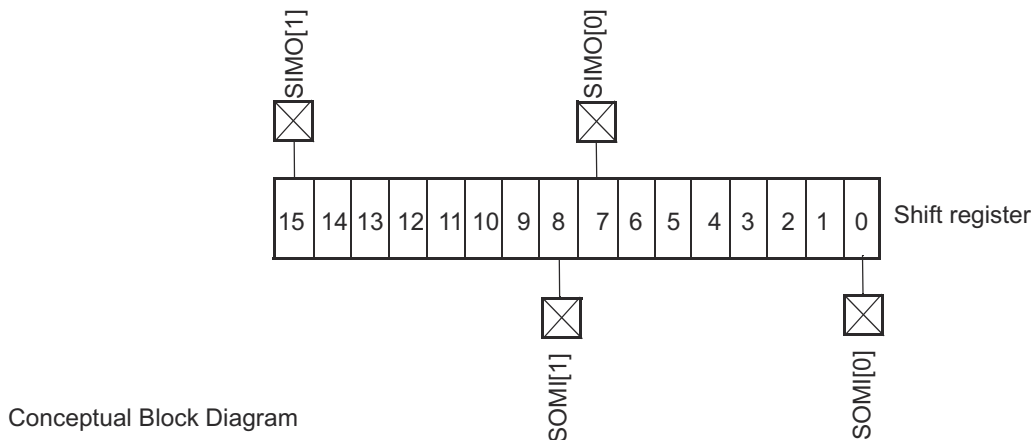
Parallel Mode	Shift Register Bit	PICO[7:0]
1	0	0
2	8	1
	0	0
4	12	3
	8	2
	4	1
	0	0
8	14	7
	12	6
	10	5
	8	4
	6	3
	4	2
	2	1
	0	0

**Table 12-6. Pin Mapping for POCI Pin with LSB First**

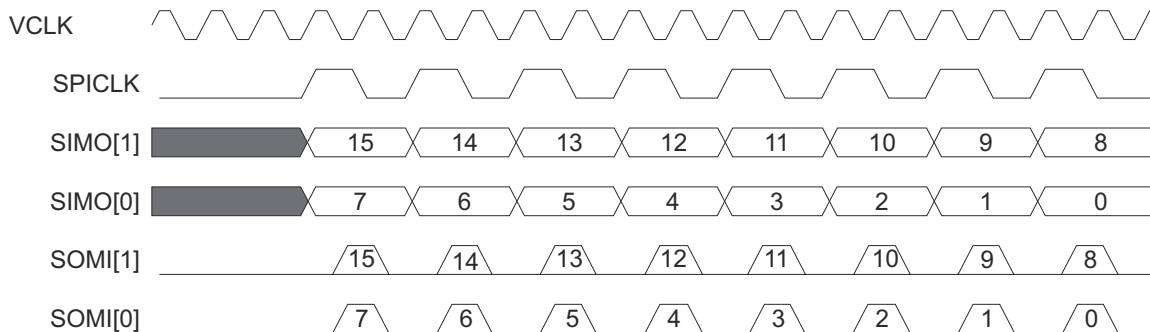
Parallel Mode	Shift Register Bit	POCI[7:0]
1	15	0
2	7	0
	15	1
4	3	0
	7	1
	11	2
	15	3
8	1	0
	3	1
	5	2
	7	3
	9	4
	11	5
	13	6
	15	7

#### 12.2.15.4 2-Data Line Mode (MSB First, Phase 0, Polarity 0)

In 2-data line mode (controller mode) the shift register bits 15 and 7 will be connected to the pins PICO[1] and PICO[0], and the shift register bits 8 and 0 will be connected to the pins POCI[1] and POCI[0] or vice versa in peripheral mode. After writing to the SPIDAT0/SPIDAT1 register, the bits 15 and 7 will be output on PICO[1] and PICO[0] on the rising edge of SPICLK. With the falling clock edge of the SPICLK, the received data on POCI[1] and POCI[0] will be latched to the shift register bits 8 and 0. The subsequent rising edge of SPICLK will shift the data in the shift register by 1 bit to the left. (PICO[1] will shift the data out from bit 15 to 8, PICO[0] will shift the data out from bit 7 to 0). After eight SPICLK cycles, when the full data word is transferred, the shift register (16 bits) is copied to the receive buffer, and the RXINT flag will be set. Figure 12-16 shows the clock/data diagram of the 2-data line mode. Figure 12-17 shows the timing of a two-pin parallel transfer.



**Figure 12-16. 2-data Line Mode (Phase 0, Polarity 0)**

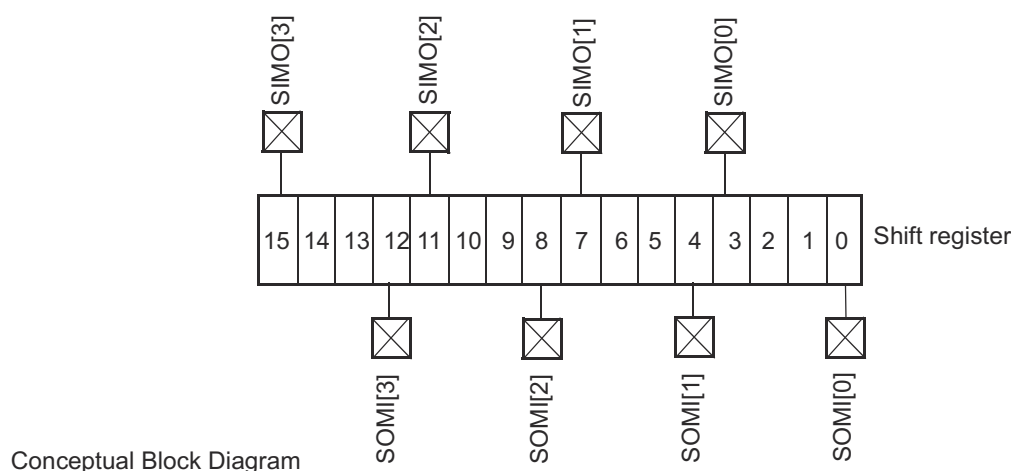


**Figure 12-17. Two-Pin Parallel Mode Timing Diagram (Phase 0, Polarity 0)**

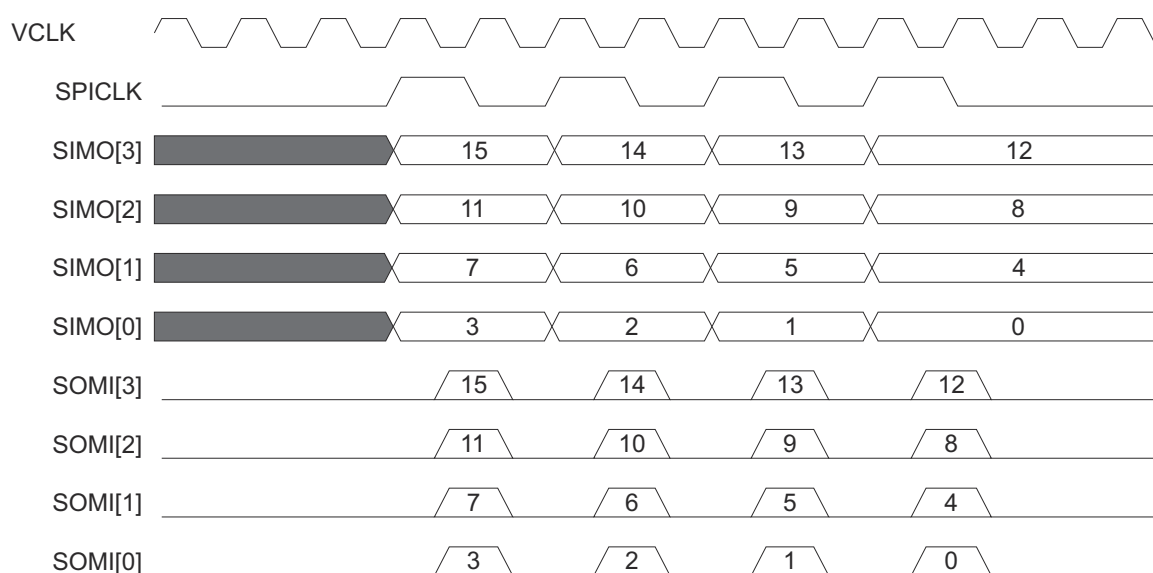
### 12.2.15.5 4-Data Line Mode (MSB First, Phase 0, Polarity 0)

In 4-data line mode (controller mode) the shift register bits 15, 11, 7, and 3 will be connected to the pins SIMO[3], SIMO[2], SIMO[1], and SIMO[0], and the shift register bits 12, 8, 4, and 0 will be connected to the pins SOMI[3], SOMI[2], SOMI[1], and SOMI[0] (or vice versa in peripheral mode). After writing to SPIDAT1/SPIDAT0, the bits 15, 11, 7, and 3 will be output on SIMO[3], SIMO[2], SIMO[1], and SIMO[0] on the rising edge of SPICLK. With the falling clock edge of the SPICLK, the received data on SOMI[3], SOMI[2], SOMI[1] and SOMI[0] will be latched to shift register bits 12, 8, 4, and 0. The subsequent rising edge of SPICLK will shift data in the shift register by 1 bit to the left (SIMO[3] will shift the data out from bit 15 to 12, SIMO[2] will shift the data out from bit 11 to 8, SIMO[1] will shift the data out from bit 7 to 4, SIMO[0] will shift the data out from bit 3 to 0). After four SPICLK cycles, when the full data word is transferred, the shift register (16 bits) is copied to the receive buffer, and the RXINT flag will be set.

Figure 12-18 shows the clock/data diagram of the four-data line mode. Figure 12-19, shows the timing diagram for four-data line mode.



**Figure 12-18. 4-Data Line Mode (Phase 0, Polarity 0)**



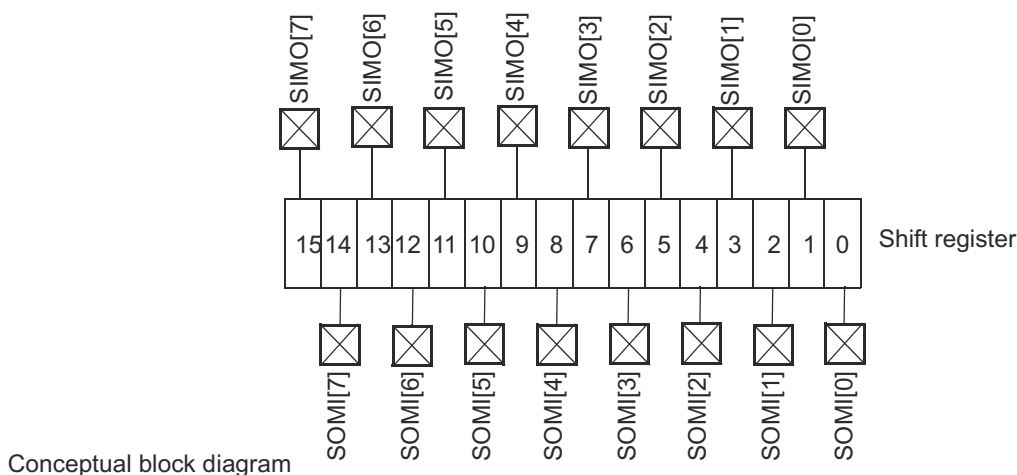
**Figure 12-19. 4 Pins Parallel Mode Timing Diagram (Phase 0, Polarity 0)**

### 12.2.15.6 8-Data Line Mode (MSB First, Phase 0, Polarity 0)

In 8-data line mode (controller mode) the shift register bits 15, 13, 11, 9, 7, 5 and 3 will be connected to the pins PICO[7], PICO[6], PICO[5], PICO[4], PICO[3], PICO[2], PICO[1], and PICO[0], and the shift-register bits 14, 12, 10, 8, 6, 4, and 0 will be connected to the pins POIC[7], POIC[6], POIC[5], POIC[4], POIC[3], POIC[2], POIC[1], and POIC[0] (or vice versa in peripheral mode).

After writing to SPIDAT0/SPIDAT1, the bits 15, 13, 11, 9, 7, 5, 3, and 1 will be output on PICO[7], PICO[6], PICO[5], PICO[4], PICO[3], PICO[2], PICO[1], and PICO[0], on the rising edge of SPICLK. On the falling clock edge of the SPICLK, the received data on POIC[8], POIC[7], POIC[6], POIC[5], POIC[4], POIC[3], POIC[2], POIC[1], and POIC[0] will be latched to the shift register bits 14, 12, 10, 8, 6, 4, 2, and 0.

The subsequent rising edge of SPICLK will shift the data in the shift register by 1 bit to the left. After two SPICLK cycles, when the full data word is transferred the shift register (16 bits) is copied to the receive buffer, and the RXINT flag will be set. Figure 12-20 shows the clock/data diagram of the 8-data line mode. Figure 12-21 shows the pin timings for 8-data line mode.

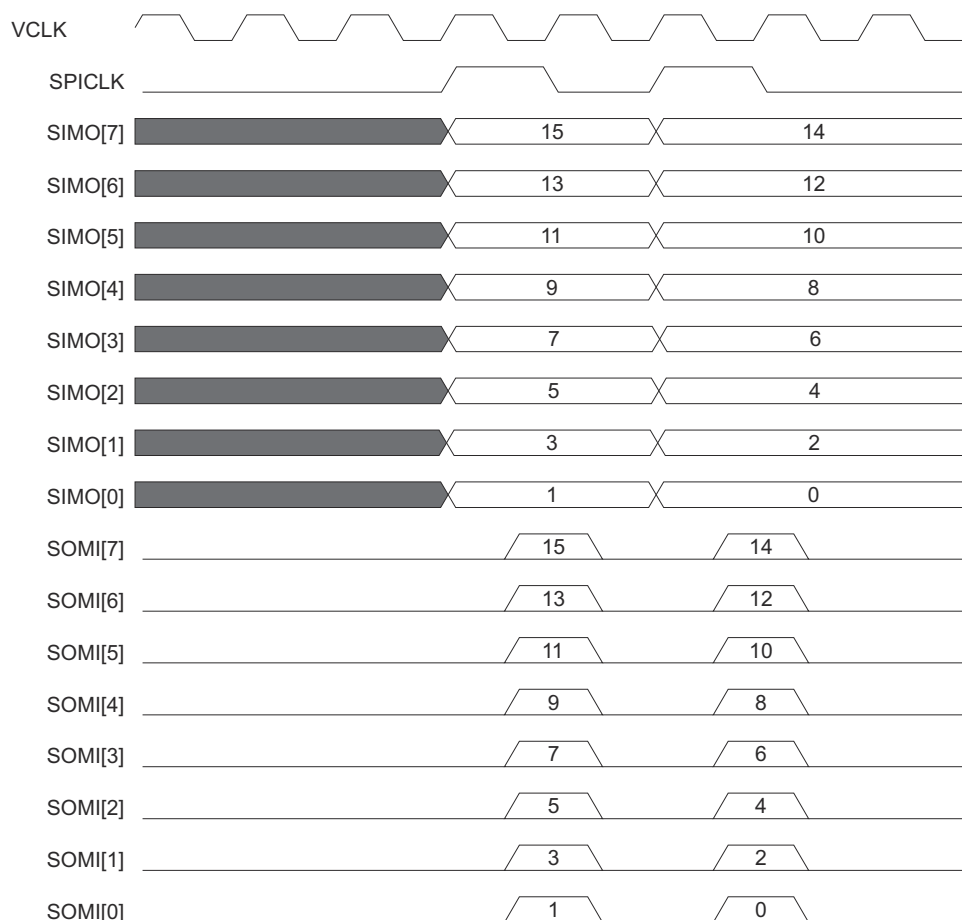


**Figure 12-20. 8-data Line Mode (Phase 0, Polarity 0)**

#### Note

##### Parity Support

Using the parity support in parallel mode may seriously affect throughput. For an eight-line mode to transfer 16 bits of data, only two SPICLK pulses are enough. If parity is enabled, one extra SPICLK pulse will be used to transfer and receive the parity bit. Parity will be transmitted and received on the 0th line regardless of 1/2/4/8-line modes. During the parity bit transfer, other data bits are not valid.



**Figure 12-21. 8 Pins Parallel Mode Timing Diagram (Phase 0, Polarity 0)**

### Note

Modulo Count Parallel Mode is not supported in this device.



### 12.2.16 Continuous Self-Test (Controller/Peripheral)

During data transfer, the SPI compares its own internal transmit data with its transmit data on the bus. The sample point for the compare is at one-half SPI clock after transmit point. If the data on the bus does not match the expected value, the bit-error (BITERR) flag is set and an interrupt is asserted if enabled.

---

#### Note

The compare is made from the output pin using its input buffer.

---

### 12.2.17 Half Duplex Mode

SPI by protocol is Full Duplex in nature, which means simultaneous TX and RX operations happen on two separate data pins, SIMO and SOMI. However, it is possible to use SPI/MibSPI to do the TX-only operation (ignoring the RX data) and the RX-only operation (using dummy TX data and ignoring the TX pin). But this requires that both SOMI and SIMO lines are bonded out in a chip to be able to support both TX-only or RX-only features.

#### 12.2.17.1 Half Duplex Mode in Master

The Half Duplex Mode gives an additional flexibility to use the SIMO pin, which is normally used as a TX pin in Controller mode, to work like an RX pin while the HDUPLEX\_ENAx bit in SPIFMTx register is set to 1. In Half Duplex Controller mode, the SIMO pin acts as an RX pin. Switching between Full Duplex and Half Duplex can be achieved using the SPIFMTx register being selected using the DFSEL bit of SPIDAT1 register or TXRAM locations.

#### 12.2.17.2 Half Duplex Mode in Peripheral

In Half Duplex Peripheral mode, the SIMO pin, which is normally an RX pin, acts as a TX pin while the HDUPLEX\_ENAx bit in SPIFMTx register is set to 1. In Half Duplex Peripheral mode, the SIMO pin acts as a TX pin. Switching between Full Duplex and Half Duplex can be achieved using the SPIFMTx register being selected using the DFSEL bit of SPIDAT1 register or TXRAM locations.

## 12.3 Test Features

### 12.3.1 Internal Loop-Back Test Mode (Controller Only)

The internal loop-back self-test mode can be utilized to test the SPI transmit and receive paths, including the shift registers, the SPI buffer registers, and the parity generator. In this mode the transmit signal is internally feedback to the receiver, whereas the SIMO, SOMI, and CLK pin are disconnected; that is, the transmitted data is internally transferred to the corresponding receive buffer while external signals remain unchanged.

This mode allows the CPU to write into the transmit buffer, and check that the receive buffer contains the correct transmit data. If an error occurs the corresponding error is set within the status field.

---

#### Note

This mode cannot be changed during transmission.

---

### 12.3.2 Input/Output Loopback Test Mode

Input/Output Loopback Test mode supports the testing of all Input/Output pins without the aid of an external interface. Loopback can be configured as either analog-loopback (loopback through the pin-level input/output buffers) or digital loopback (internal to the SPI module). With Input/Output Loopback, all functional features of the SPI can be tested. Transmit data is fed back through the receive-data line(s). See Figure 12-22 for a diagram of the types of feedback available. The IOLPBKTSTCR register defines all of the available control fields.

In loopback mode, it is also possible to induce various error conditions. See Section 12.8.44 for details of the register field controlling these features.

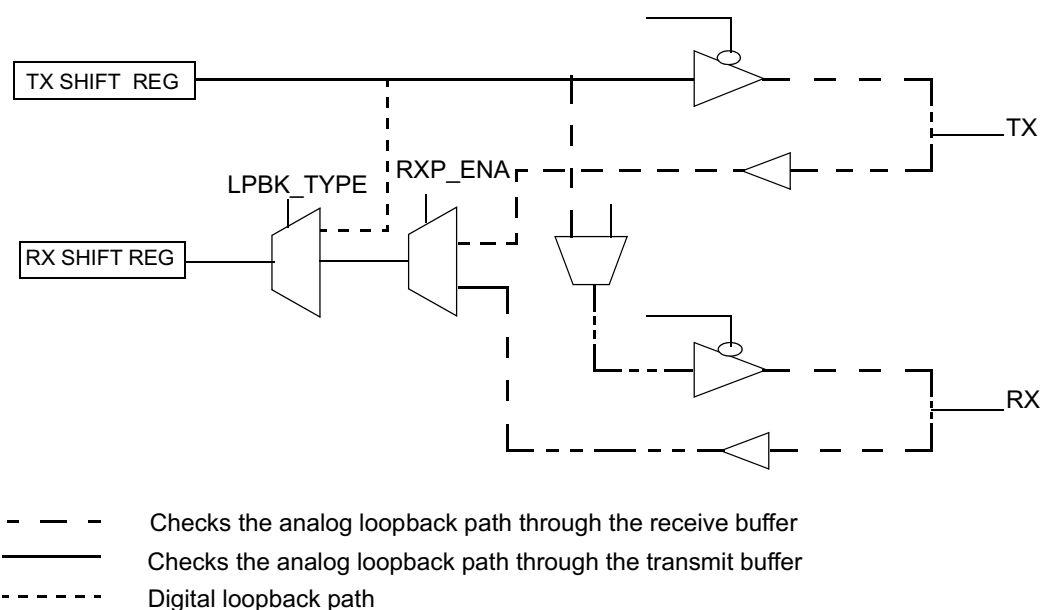
In Input/Output loopback test modes, even when the module is in slave mode, the SPICLK is generated internally. This SPICLK is used for all loopback-mode SPI transactions. Peripheral-mode features can be tested without the help of another controller SPI, using the internally-generated SPICLK. Chip selects are also generated by the slave itself while it is in Input/Output loopback mode.

In Input/Output loopback test modes, if the module is in controller mode, the  $\overline{\text{ENA}}$  signal is also generated by internal logic so that an external interface is not required.

#### Note

##### Usage Guideline for Input/Output Loopback

Input/Output Loopback mode should be used with caution because, in some configurations, even the receive pins will be driven with transmit data. During testing, it should be ensured that none of the SPI pins are driven by any other device connected to them. Otherwise, if analog loopback is selected in I/O Loopback mode, then testing may damage the device.



This diagram is intended to illustrate loopback paths and therefore may omit some normal-mode paths.

**Figure 12-22. I/O Paths during I/O Loopback Modes**

### 12.3.2.1 Input/Output Loopback Mode Operation in Peripheral Mode

In multi-buffer slave mode, there are some additional requirements for using I/O loopback mode (IOLPBK). In multi-buffer peripheral mode, the chip-select pins are the triggers for various TGs. Enabling the IOLPBK mode by writing 0xA to the IOLPBTSTENA bits of the IOLPBKTSTCR register triggers TG0 by driving SPICS to 0. The actual number of chip selects can be programmed to have any or all of the SPICS pins as functional. All other configurations should be completed before enabling the IOLPBK mode in multi-buffer peripheral mode since it triggers TG0.

After the first buffer transfer is completed, the CSNR field of the current buffer is used to trigger the next buffer. So, if multiple TGs are desired to be tested, then the CSNR field of the final buffer in each TG should hold the number of the next TG to be triggered. As long as TG boundaries are well defined and are enabled, the completion of one TG will trigger the next TG.

To stop the transfer in multi-buffer peripheral mode in I/O Loopback configuration, either IOLPBK mode can be disabled by writing 0x5 to the IOLPBTSTENA bits or all of the TGs can be disabled.

## 12.4 General-Purpose I/O

All of the SPI pins may be programmed via the SPIPCx control registers to be either functional or general-purpose I/O pins.

If the SPI function is to be used, application software must ensure that at least the SPICLK pin and the SOMI and/or SIMO pins are configured as SPI functional pins, and not as GIO pins, or else the SPI state machine will be held in reset, preventing SPI transactions.

SPI pins support:

- internal pull-up resistors
- internal pull-down resistors
- open-drain or push-pull mode

## 12.5 Interrupts

There are two levels of vectorized interrupts supported by the SPI. These interrupts can be caused under the following circumstances:

- Transmission error
- Receive overrun
- Receive complete (receive buffer full)
- Transmit buffer empty

These interrupts may be enabled or disabled via the SPIINT0 register.

During transmission, if one of the following errors occurs: BITERR, DESYNC, DLENERR, PARITYERR, or TIMEOUT, the corresponding bit in the SPIFLG register is set. If the corresponding enable bit is set, then an interrupt is generated. The level of all the above interrupts is set by the bit fields in the SPILVL register.

The error interrupts are enabled and prioritized independently from each other, but the interrupt generated will be the same if multiple errors are enabled on the same level. The SPIFLG register should be used to determine the actual cause of an error.

---

### Note

Since there are two interrupt lines, one each for Level 0 and Level 1, it is possible for a programmer to separate out the interrupts for receive buffer full and transmit buffer empty. By programming one to Level 0 and the other to Level 1, it is possible to avoid a check on whether an interrupt occurred for transmit or for receive. A programmer can also choose to group all of the error interrupts into one interrupt line and both TX-empty and RX-full interrupts into another interrupt line using the LVL control register. In this way, it is possible to separate error-checking from normal data handling.

---

### 12.5.1 Interrupts in Multi-Buffer Mode

In multi-buffer mode, the SPI can generate interrupts on two levels.

In normal multi-buffer operation, the receive and transmit are not used and therefore the enable bits of SPIINT0 are not used.

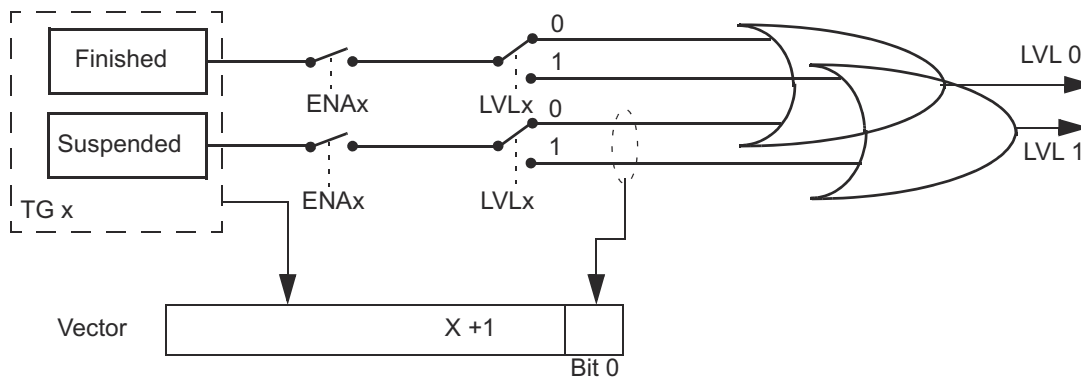
The interrupts available in multi-buffer mode are:

- Transmission error interrupt
- Receive overrun interrupt
- TG suspended interrupt
- TG completed interrupt

When a TG has finished and the corresponding enable bit in the TGINTENA register is set, a transfer-finished interrupt is generated. The level of priority of the interrupt is determined by the corresponding bit in the TGINTLVL register.

When a TG is suspended by a buffer that has been set as suspend to wait until TXFULL flag or/and RXEMPTY flag are set, and if the corresponding bit in the TGINTENA register is set, a transfer-suspended interrupt is generated. The level of priority of the interrupt is determined by the corresponding bit in the TGINTLVL register.

[Figure 12-23](#) illustrates the TG interrupts.

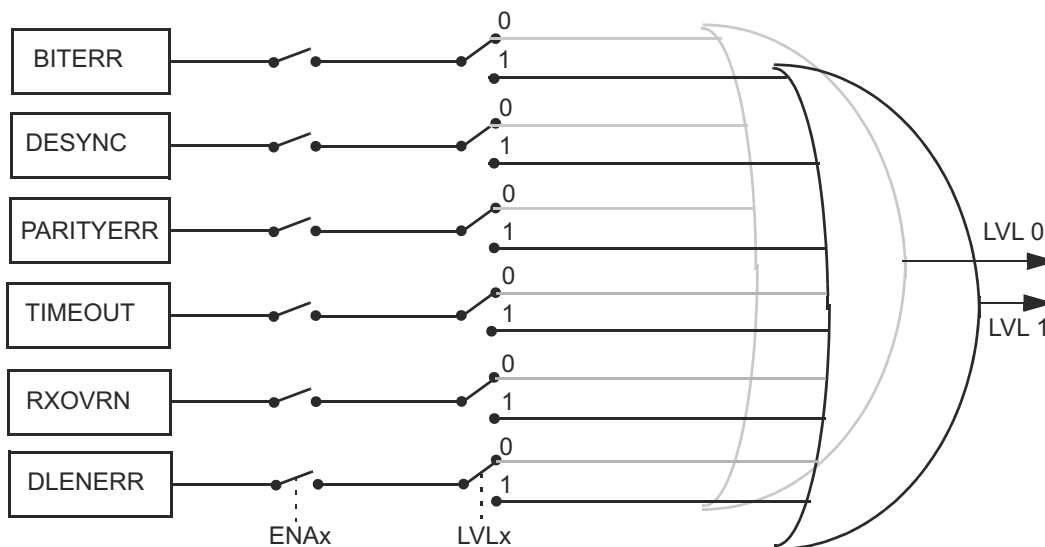


**Figure 12-23. TG Interrupt Structure**

During transmission, if one of the following errors occurs, BITERR, DESYNC, PARITYERR, TIMEOUT, DLENERR, the corresponding flag in the SPIFLG register is set. If the enable bit is set, then an interrupt is generated. The level of the interrupts could be generated according to the bit field in SPILVL register.

The RXOVRN interrupt is generated when a buffer in the RXRAM is overwritten by a new received word. While writing newly received data to a RXRAM location, if the RXEMPTY bit of the corresponding location is 0, then the RXOVR bit will be set to 1 during the write operation, so that the buffer starts to indicate an overrun. This RXOVR flag is also reflected in SPIFLG register as RXOVRNINTFLG and the corresponding vector number is updated in TGINTVECT0/TGINTVECT1 register. If an overrun interrupt is enabled, then an interrupt will be generated indicating an overrun condition.

The error interrupts are enabled and prioritized independently from each other, but the vector generated by the SPI will be the same if multiple errors are enabled on the same level.



**Figure 12-24. SPIFLG Interrupt Structure**

Since the priority of an error interrupt is lower than a completion/suspend interrupt for a TG, the interrupts can be split into two levels. By programming all the error interrupts into Level 0 and TG-complete / TG-suspend interrupts into Level 1, it is possible to get a clear indication of the source of error interrupts. However, when a vector register shows an error interrupt, the actual buffer for which the error has occurred is not readily identifiable. Since each buffer in the multi-buffer RAM is stored along with its individual status flags, each buffer should be read until a buffer with any error flag set is found.

A separate interrupt line is provided to indicate the uncorrectable error condition in the MibSPI. This line is available (and valid) only in the multi-buffer mode of the MibSPI module and if the parity error detection feature for multi-buffer RAM is enabled.

## 12.6 DMA Interface

In order to reduce CPU overhead in handling SPI message traffic on a character-by-character basis, SPI can use the DMA controller to transfer the data. The DMA request enable bit (DMA REQ EN) controls the assertion of requests to the DMA controller module. When a character is being transmitted or received, the SPI will signal the DMA via the DMA request signals, TX\_DMA\_REQ and RX\_DMA\_REQ. The DMA controller will then perform the required data transfer.

For efficient behavior during DMA operations, the transmitter empty and receive-buffer full interrupts can be disabled. For specific DMA features, see the DMA controller specification.

The SPI generates a request on the TX\_DMA\_REQ line each time the TX data is copied to the TX shift register either from the TXBUF or from peripheral data bus (when TXBUF is empty).

The first TX\_DMA\_REQ pulse is generated when either of the following is true:

- DMAREQEN (SPIINT0[16]) is set to 1 while SPIEN (SPIGCR1[24]) is already 1.
- SPIEN (SPIGCR1[24]) is set to 1 while DMAREQEN (SPIINT0[16]) is already 1.

The SPI generates a request on the RX\_DMA\_REQ line each time the received data is copied to the SPIBUF.

### 12.6.1 DMA in Multi-Buffer Mode

The MibSPI provides sophisticated programmable DMA control logic that completely eliminates the necessity of CPU intervention for data transfers, once programmed. When the multi-buffer mode is used, the DMA enable bit in the SPIINT0 register is ignored. DMA source or destination should be only the multi-buffer RAM and not SPIDAT0 / SPIDAT1 or SPIBUF register as in case of compatibility mode DMA.

The MibSPI offers up to eight DMA channels (for SEND and RECEIVE). All of the DMA channels are programmable individually and can be hooked to any buffer. The MibSPI provides up to 16 DMA request lines, and DMA requests from any channel can be programmed to be routed through any of these 16 lines. A DMA transfer can trigger both transmit and receive.

Each DMA channel has the capability to transfer a block of up to 32 data words without interruption using only one buffer of the array by configuring the DMAxCTRL register. Using the DMAxCOUNT and DMAxTNTLEN register, up to 65535 (64K) words of data can be transferred without any interruption using just one buffer of the array. This enables the transfer of memory blocks from or into an external SPI memory.

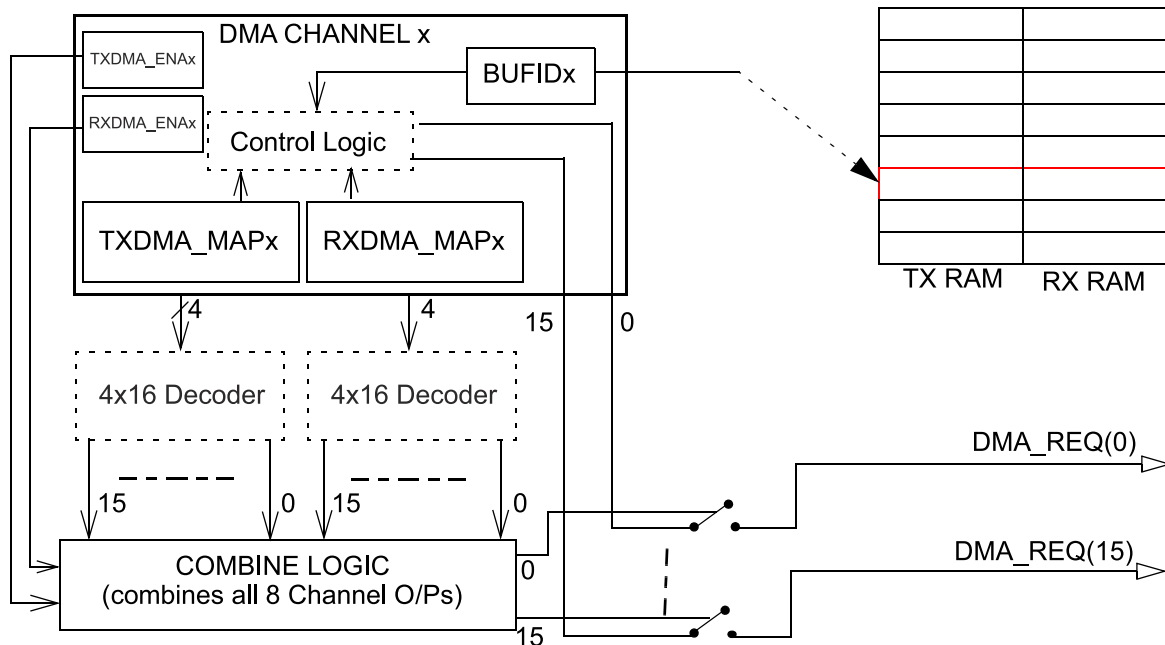


Figure 12-25. DMA Channel and Request Line (Logical) Structure in Multi-buffer Mode

## 12.7 Module Configuration

MibSPI/MibSPIP can be configured to function as Normal SPI and Multi-buffered SPI. Upon power-up or a system-level reset, each bit in the module registers is set to a default state. The registers are writable only after the RESET bit is set to 1.

### 12.7.1 Compatibility (SPI) Mode Configuration

The following list details the configuration steps that software should perform prior to the transmission or reception of data. As long as the SPIEN bit in the Global Control Register 1 (SPIGCR1) is cleared to 0 the entire time that the SPI is being configured, the order in which the registers are programmed is not important.

- Enable SPI by setting RESET bit.
- Configure the SIMO, SOMI, SPICLK, and optional  $\overline{\text{SPICS}}$  and  $\overline{\text{SPIENA}}$  pins for SPI functionality by setting the corresponding bit in SPIPC0 register.
- Configure the module to function as Controller or Peripheral using CLKMOD and MASTER bits.
- Configure the required SPI data format using SPIFMTx register.
- If the module is selected to function as Controller, the delay parameters can be configured using SPIDELAY register.
- Enable the Interrupts using SPIINT0 register if required.
- Select the chip select to be used by setting CSNR bits in SPIDAT1 register.
- Configure CSHOLD and WDEL bits in SPIDAT1 register if required.
- Select the Data word format by setting DFSEL bits. Select the Number of the configured SPIFMTx register (0 to 3) to used for the communication.
- Set LOOPBACK bit to connect the transmitter to the receiver internally. (This feature is used to perform a self-test. Do not configure for normal communication to external devices).
- Set SPIEN bit to 1 after the SPI is configured.
- Perform Transmit and receive data, using SPIDAT1 and SPIBUF register.
- You must wait for TXFULL to reset or TXINT before writing next data to SPIDAT1 register.
- You must wait for RXEMPTY to reset or RXINT before reading the data from SPIBUF register.



### 12.7.2 MibSPI Mode Configuration

The following list details the configuration steps that software should perform prior to the transmission or reception of data in MIBSPI mode. As long as the SPIEN bit in the Global Control Register 1 (SPIGCR1) is cleared to 0 the entire time that the SPI is being configured, the order in which the registers are programmed is not important.

- Enable SPI by setting RESET bit.
- Set MSPIENA bit to 1 to get access to multi-buffer mode registers.
- Configure the SIMO, SOMI, SPICLK, and optional  $\overline{\text{SPICS}}$  and  $\overline{\text{SPIENA}}$  pins for SPI functionality by setting the corresponding bit in SPIPC0 register.
- Configure the module to function as Controller or Peripheral using CLKMOD and MASTER bits.
- Configure the required SPI data format using SPIFMTx register.
- If the module is selected to function as Controller, the delay parameters can be configured using SPIDELAY register.
- Check for BUFINITACTIVE bit to be active before configuring MIBSPI RAM. (From Device Power On it take Number of Buffers × Peripheral clock period to initialize complete RAM.)
- Enable the Transfer Group interrupts using TGITENST register if required.
- Enable error interrupts using SPIINT0 register if required.
- Set SPIEN bit to 1 after the SPI is configured.
- The Trigger Source, Trigger Event, Transfer Group start address for the corresponding Transfer groups can be configured using the corresponding TGxCTRL register.
- Configure LPEND to specify the end address of the last TG.
- Similar to SPIDAT1 register, the 16 bit control fields in every TXRAM buffer in the TG have to be configured.
- Configure one of the eight BUFMODE available for each buffer.
- Fill the data to be transmitted in TXDATA field in TXRAM buffers.
- Configure TGENA bit to enable the required Transfer groups. (In case of Trigger event always setting TGENA will trigger the transfer group).
- At the occurrence of the correct trigger event, the Transfer group will be triggered and data gets transmitted and received one after the other with out any CPU intervention.
- You can poll Transfer Group interrupt flag or wait for a transfer-completed interrupt to read and write new data to the buffers.

## 12.8 Control Registers

This section describes the SPI control, data, and pin registers. The registers support 8-bit, 16-bit and 32-bit writes. The offset is relative to the associated base address of this module in a system.

### Note

TI highly recommends that write values corresponding to the reserved locations of registers be maintained as 0 consistently. This allows future enhancements to use these reserved bits as control bits without affecting the functionality of the module with any older versions of software.

**Table 12-7. SPI Registers**

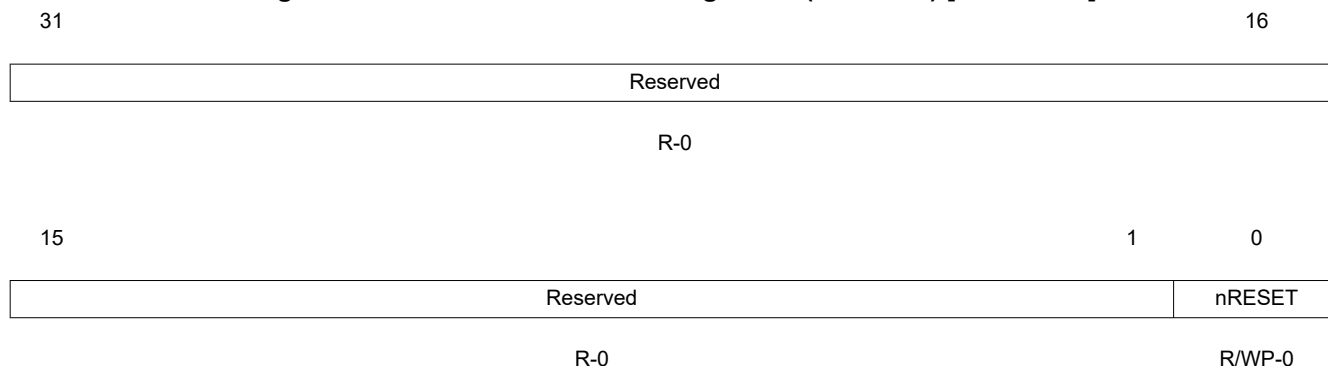
Offset	Acronym	Register Description	Section
00h	SPIGCR0	SPI Global Control Register 0	<a href="#">Section 12.8.1</a>
04h	SPIGCR1	SPI Global Control Register 1	<a href="#">Section 12.8.2</a>
08h	SPIINT0	SPI Interrupt Register	<a href="#">Section 12.8.3</a>
0Ch	SPILV	SPI Interrupt Level Register	<a href="#">Section 12.8.4</a>
10h	SPIFLG	SPI Flag Register	<a href="#">Section 12.8.5</a>
14h	SPIPC0	SPI Pin Control Register 0	<a href="#">Section 12.8.6</a>
18h	SPIPC1	SPI Pin Control Register 1	<a href="#">Section 12.8.7</a>
1Ch	SPIPC2	SPI Pin Control Register 2	<a href="#">Section 12.8.8</a>
20h	SPIPC3	SPI Pin Control Register 3	<a href="#">Section 12.8.9</a>
24h	SPIPC4	SPI Pin Control Register 4	<a href="#">Section 12.8.10</a>
28h	SPIPC5	SPI Pin Control Register 5	<a href="#">Section 12.8.11</a>
2Ch	SPIPC6	SPI Pin Control Register 6	<a href="#">Section 12.8.12</a>
30h	SPIPC7	SPI Pin Control Register 7	<a href="#">Section 12.8.13</a>
34h	SPIPC8	SPI Pin Control Register 8	<a href="#">Section 12.8.14</a>
38h	SPIDAT0	SPI Transmit Data Register 0	<a href="#">Section 12.8.15</a>
3Ch	SPIDAT1	SPI Transmit Data Register 1	<a href="#">Section 12.8.16</a>
40h	SPIBUF	SPI Receive Buffer Register	<a href="#">Section 12.8.17</a>
44h	SPIEMU	SPI Emulation Register	<a href="#">Section 12.8.18</a>
48h	SPIDELAY	SPI Delay Register	<a href="#">Section 12.8.19</a>
4Ch	SPIDEF	SPI Default Chip Select Register	<a href="#">Section 12.8.20</a>
50h-5Ch	SPIFMT0-SPIFMT3	SPI Data Format Registers	<a href="#">Section 12.8.21</a>
60h	INTVECT0	Interrupt Vector 0	<a href="#">Section 12.8.22</a>
64h	INTVECT1	Interrupt Vector 1	<a href="#">Section 12.8.24</a>
68h	SPIPC9 <sup>(1)</sup>	SPI Pin Control Register 9	<a href="#">Section 12.8.25</a>
6Ch	SPIPMCTRL	Parallel/Modulo Mode Control Register	<a href="#">Section 12.8.26</a>
70h	MIBSPIE	Multi-buffer Mode Enable Register	<a href="#">Section 12.8.27</a>
74h	TGINTENST	TG Interrupt Enable Set Register	<a href="#">Section 12.8.28</a>
78h	TGINTENCR	TG Interrupt Enable Clear Register	<a href="#">Section 12.8.29</a>
7Ch	TGITLVST	Transfer Group Interrupt Level Set Register	<a href="#">Section 12.8.30</a>
80h	TGITLVCR	Transfer Group Interrupt Level Clear Register	<a href="#">Section 12.8.31</a>
84h	TGINTFLG	Transfer Group Interrupt Flag Register	<a href="#">Section 12.8.32</a>
88h-8Ch	Reserved	Reserved	
90h	TICKCNT	Tick Count Register	<a href="#">Section 12.8.33</a>
94h	LTGPEND	Last TG End Pointer	<a href="#">Section 12.8.34</a>
98h-D4h	TGxCTRL	TGx Control Registers	<a href="#">Section 12.8.35</a>
D8h-F4h	DMAxCTRL	DMA Channel Control Register	<a href="#">Section 12.8.36</a>
F8h-114h	ICOUNT	DMAxCOUNT Register	<a href="#">Section 12.8.37</a>

**Table 12-7. SPI Registers (continued)**

Offset	Acronym	Register Description	Section
118h	DMACNTLEN	DMA Large Count	<a href="#">Section 12.8.38</a>
11Ch	Reserved	Reserved	
120h	UERRCTRL	Multi-buffer RAM Uncorrectable Parity Error Control Register	<a href="#">Section 12.8.39</a>
124h	UERRSTAT	Multi-buffer RAM Uncorrectable Parity Error Status Register	<a href="#">Section 12.8.40</a>
128h	UERRADDR1	RXRAM Uncorrectable Parity Error Address Register	<a href="#">Section 12.8.41</a>
12Ch	UERRADDR0	TXRAM Uncorrectable Parity Error Address Register	<a href="#">Section 12.8.42</a>
130h	RXOVRN_BUF_ADDR	RXRAM Overrun Buffer Address Register	<a href="#">Section 12.8.43</a>
134h	IOLPBKTSTCR	I/O Loopback Test Control Register	<a href="#">Section 12.8.44</a>
138h	EXTENDED_PRESCALE1	SPI Extended Prescale Register 1	<a href="#">Section 12.8.45</a>
13Ch	EXTENDED_PRESCALE2	SPI Extended Prescale Register 2	<a href="#">Section 12.8.46</a>

(1) SPIPC9 only applies to SPI2.

### 12.8.1 SPI Global Control Register 0 (SPIGCR0)

**Figure 12-26. SPI Global Control Register 0 (SPIGCR0) [offset = 00]**


LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-8. SPI Global Control Register 0 (SPIGCR0) Field Descriptions**

Bit	Field	Value	Description
31-1	Reserved	0	Reads return 0. Writes have no effect.
0	nRESET	0	This is the local reset control for the module. This bit needs to be set to 1 before any operation on SPI / MibSPI can be done. Only after setting this bit to 1, the Auto Initialization of Multi-buffer RAM starts. Clearing this bit to 0 will result in all of the control and status register values to return to their default values..
		1	SPI is in the reset state.
		1	SPI is out of the reset state.

## 12.8.2 SPI Global Control Register 1 (SPIGCR1)

**Figure 12-27. SPI Global Control Register 1 (SPIGCR1) [offset = 04h]**

31		25	24	23		17		16
Reserved			SPIEN	Reserved			LOOPBACK	
R-0			R/W-0	R-0			R/WP-0	
15		9	8	7		2	1	0
Reserved			POWERDOWN	Reserved			CLKMOD	MASTER
R-0			R/W-0	R-0			R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-9. SPI Global Control Register 1 (SPIGCR1) Field Descriptions**

Bit	Field	Value	Description
31-25	Reserved	0	Reads return 0. Writes have no effect.
24	SPIEN	0 1	SPI enable. This bit enables SPI transfers. This bit must be set to 1 after all other SPI configuration bits have been written. When the SPIEN bit is 0 or cleared to 0, the following SPI registers get forced to their default states: <ul style="list-style-type: none"> <li>Both TX and RX shift registers</li> <li>The TXDATA fields of the SPI Transmit Data Register 0 (SPIDAT0) and the SPI Transmit Data Register 1 (SPIDAT1)</li> <li>All the fields of the SPI Flag Register (SPIFLG)</li> <li>Contents of SPIBUF and the internal RXBUF registers</li> </ul> The SPI is not activated for transfers. Activates SPI.
23-17	Reserved	0	Reads return 0. Writes have no effect.
16	LOOPBACK	0 1	Internal loop-back test mode. The internal self-test option can be enabled by setting this bit. If the SPISIMO and SPISOMI pins are configured with SPI functionality, then the SPISIMO[7:0] pins are internally connected to the SPISOMI[7:0] pins (transmit data is looped back as receive data). GIO mode for these pins is not supported in loopback mode. Externally, during loop-back operation, the SPICLK pin outputs an inactive value and SPISOMI[7:0] remains in the high-impedance state. If the SPI is initialized in slave mode or a data transfer is ongoing, errors may result.  <b>Note: This loopback mode can only be used in master mode. Master mode must be selected before setting LOOPBACK. When this mode is selected, the CLKMOD bit should be set to 1, meaning that SPICLK is internally generated.</b> Internal loop-back test mode is disabled. Internal loop-back test mode is enabled.
15-9	Reserved	0	Reads return 0. Writes have no effect.
8	POWERDOWN	0 1	When active, the SPI state machine enters a power-down state. The SPI is in active mode. The SPI is in power-down mode.
7-2	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-9. SPI Global Control Register 1 (SPIGCR1) Field Descriptions (continued)**

Bit	Field	Value	Description
1	CLKMOD	0 1	Clock mode. This bit selects either an internal or external clock source. This bit also determines the I/O direction of the SPIENA and SPICS pins in functional mode.  Clock is external. <ul style="list-style-type: none"> <li>SPIENA is an output.</li> <li>SPICS are inputs.</li> </ul> Clock is internally-generated. <ul style="list-style-type: none"> <li>SPIENA is an input.</li> <li>SPICS are outputs.</li> </ul>
0	MASTER	0 1	SPISIMO/SPISOMI pin direction determination. Sets the direction of the SPISIMO and SPISOMI pins.  <b>Note: For master-mode operation of the SPI, MASTER bit should be set to 1 and CLKMOD bit can be set either 1 or 0. The master-mode SPI can run on an external clock on SPICLK.</b>  <b>For slave mode operation, both the MASTER and CLKMOD bits should be cleared to 0. Any other combinations may result in unpredictable behavior of the SPI. In slave mode, SPICLK will not be generated internally in slave mode.</b>  0 SPISIMO[7:0] pins are inputs, SPISOMI[7:0] pins are outputs. 1 SPISOMI[7:0] pins are inputs, SPISIMO[7:0] pins are outputs.

### 12.8.3 SPI Interrupt Register (SPIINT0)

**Figure 12-28. SPI Interrupt Register (SPIINT0) [offset = 08h]**

31						25	24
Reserved						ENABLEHIGHZ	
R-0						R/W-0	
23						17	16
Reserved						DMAREQEN	
R-0						R/W-0	
15					10	9	8
Reserved						TXINT ENA	RXINT ENA
R-0						R/W-0	R/W-0
7	6	5	4	3	2	1	0
Reserved	RXOVRNINT ENA	Reserved	BITERR ENA	DESYNC ENA	PARERR ENA	TIMEOUT ENA	DLENERR ENA
R-0	R/W-0	R-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-10. SPI Interrupt Register (SPIINT0) Field Descriptions**

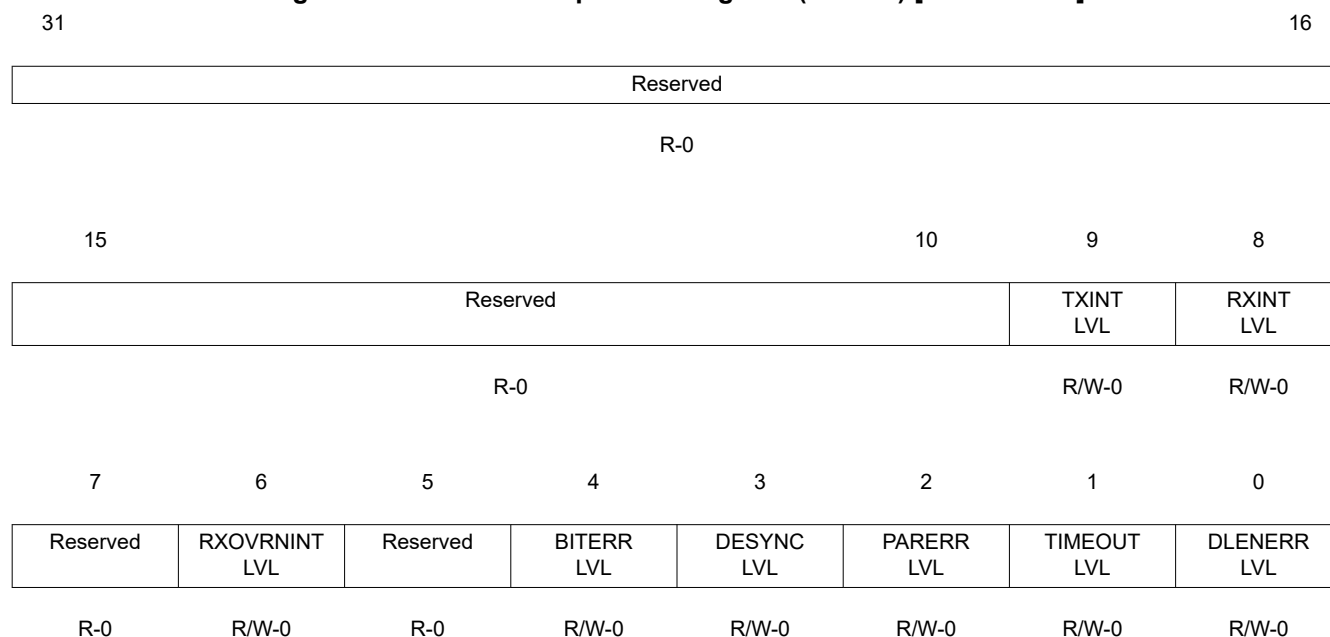
Bit	Field	Value	Description
31-25	Reserved	0	Reads return 0. Writes have no effect.
24	ENABLEHIGHZ	0 1	<p>SPIEN<math>\overline{A}</math> pin high-impedance enable. When active, the SPIEN<math>\overline{A}</math> pin (when it is configured as a WAIT functional output signal in a slave SPI) is forced to high-impedance when not driving a low signal. If inactive, then the pin will output both a high and a low signal.</p> <p>0 SPIEN<math>\overline{A}</math> pin is pulled high when not active.</p> <p>1 SPIEN<math>\overline{A}</math> pin remains high-impedance when not active.</p>
23-17	Reserved	0	Reads return 0. Writes have no effect.
16	DMAREQEN	0 1	<p>DMA request enable. Enables the DMA request signal to be generated for both receive and transmit channels. Enable DMA REQ only after setting the SPIEN bit to 1.</p> <p>0 DMA is not used.</p> <p>1 DMA requests will be generated.</p> <p><b>Note: A DMA request will be generated on the TX DMA REQ line each time a word is copied to the shift register either from TXBUF or directly from SPIDAT0/SPIDAT1 writes.</b></p> <p><b>Note: A DMA request will be generated on the RX DMA REQ line each time a word is copied to the SPIBUF register either from RXBUF or directly from the shift register.</b></p>
15-10	Reserved	0	Reads return 0. Writes have no effect.
9	TXINTENA	0 1	<p>Causes an interrupt to be generated every time data is written to the shift register, so that the next word can be written to TXBUF. Setting this bit will generate an interrupt if the TXINTFLG bit (SPI Flag Register (SPIFLG)[9]) is set to 1.</p> <p>0 No interrupt will be generated upon TXINTFLG being set to 1.</p> <p>1 An interrupt will be generated upon TXINTFLG being set to 1.</p> <p>The transmitter empty interrupt is valid in compatibility mode of SPI only. In multi-buffered mode, this interrupts will not be generated, even if it is enabled.</p> <p><b>Note: An interrupt request will be generated as soon as this bit is set to 1. By default it will be generated on the INT0 line. The SPILVL register can be programmed to change the interrupt line.</b></p>
8	RXINTENA	0 1	<p>Causes an interrupt to be generated when the RXINTFLAG bit (SPI Flag Register (SPIFLG)[8]) is set by hardware.</p> <p>0 Interrupt will not be generated.</p> <p>1 Interrupt will be generated.</p> <p>The receiver full interrupt is valid in compatibility mode of SPI only. In multi-buffered mode, this interrupts will not be generated, even if it is enabled.</p>
7	Reserved	0	Reads return 0. Writes have no effect.
6	RXOVRNINTENA	0 1	<p>Overrun interrupt enable.</p> <p>0 Overrun interrupt will not be generated.</p> <p>1 Overrun interrupt will be generated.</p>
5	Reserved	0	Reads return 0. Writes have no effect.
4	BITERRENA	0 1	<p>Enables interrupt on bit error.</p> <p>0 No interrupt asserted upon bit error.</p> <p>1 Enables interrupt on bit error.</p>
3	DESYNCENA	0 1	<p>Enables interrupt on desynchronized slave. DESYNCENA is used in master mode only.</p> <p>0 No interrupt asserted upon desynchronization error.</p> <p>1 An interrupt is asserted on desynchronization of the slave (DESYNC = 1).</p>
2	PARERRENA	0 1	<p>Enables interrupt-on-parity-error.</p> <p>0 No interrupt asserted on parity error.</p> <p>1 An interrupt is asserted on a parity error.</p>

**Table 12-10. SPI Interrupt Register (SPIINT0) Field Descriptions (continued)**

Bit	Field	Value	Description
1	TIMEOUTENA	0 1	Enables interrupt on ENA signal time-out. No interrupt asserted upon ENA signal time-out. An interrupt is asserted on a time-out of the ENA signal.
0	DLENERRENA	0 1	Data length error interrupt enable. A data length error occurs under the following conditions. <b>Master:</b> When $\overline{\text{SPIENA}}$ is used, if the $\overline{\text{SPIENA}}$ pin from the slave is deasserted before the master has completed its transfer, the data length error is set. That is, if the character length counter has not overflowed while $\overline{\text{SPIENA}}$ deassertion is detected, then it means that the slave has neither received full data from the master nor has it transmitted complete data. <b>Slave:</b> When $\overline{\text{SPICS}}$ pins are used, if the incoming valid $\overline{\text{SPICS}}$ pin is deactivated before the character length counter overflows, then the data length error is set. No interrupt is generated upon data length error. An interrupt is asserted when a data-length error occurs.

## 12.8.4 SPI Interrupt Level Register (SPILVL)

**Figure 12-29. SPI Interrupt Level Register (SPILVL) [offset = 0Ch]**



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-11. SPI Interrupt Level Register (SPILVL) Field Descriptions**

Bit	Field	Value	Description
31-10	Reserved	0	Reads return 0. Writes have no effect.
9	TXINTLVL	0 1	Transmit interrupt level. Transmit interrupt is mapped to interrupt line INT0. Transmit interrupt is mapped to interrupt line INT1.
8	RXINTLVL	0 1	Receive interrupt level. Receive interrupt is mapped to interrupt line INT0. Receive interrupt is mapped to interrupt line INT1.
7	Reserved	0	Reads return 0. Writes have no effect.
6	RXOVRNINTLVL	0 1	Receive overrun interrupt level. Receive overrun interrupt is mapped to interrupt line INT0. Receive overrun interrupt is mapped to interrupt line INT1.
5	Reserved	0	Reads return 0. Writes have no effect.
4	BITERRLVL	0 1	Bit error interrupt level. Bit error interrupt is mapped to interrupt line INT0. Bit error interrupt is mapped to interrupt line INT1.
3	DESYNCLVL	0 1	Desynchronized slave interrupt level. (master mode only). An interrupt caused by desynchronization of the slave is mapped to interrupt line INT0. An interrupt caused by desynchronization of the slave is mapped to interrupt line INT1.
2	PARERRLVL	0 1	Parity error interrupt level. A parity error interrupt is mapped to interrupt line INT0. A parity error interrupt is mapped to interrupt line INT1.



**Table 12-11. SPI Interrupt Level Register (SPILVL) Field Descriptions (continued)**

Bit	Field	Value	Description
1	TIMEOUTLVL	0	SPIEN $\bar{A}$ pin time-out interrupt level.
		1	An interrupt on a time-out of the ENA signal (TIMEOUT = 1) is mapped to interrupt line INT0.
0	DLENERRLVL	0	An interrupt on a time-out of the ENA signal (TIMEOUT = 1) is mapped to interrupt line INT1.
		1	Data length error interrupt level (line) select.
		0	An interrupt on data length error is mapped to interrupt line INT0.
		1	An interrupt on data length error is mapped to interrupt line INT1.

## 12.8.5 SPI Flag Register (SPIFLG)

Software must check all flag bits when reading this register.

**Figure 12-30. SPI Flag Register (SPIFLG) [offset = 10h]**

31		25	24	23			16
Reserved			BUFINIT ACTIVE	Reserved			
R-0			R-0	R-0			
15				10	9	8	
Reserved					TXINT FLG	RXINT FLG	
R-0					R-0	R/W1C-0	
7	6	5	4	3	2	1	0
Reserved	RXOVRNINT FLG	Reserved	BITERR FLG	DESYNC FLG	PARERR FLG	TIMEOUT FLG	DLENERR FLG
R-0	R/W1C-0	R-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0	R/W1C-0

LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear; -n = value after reset

**Table 12-12. SPI Flag Register (SPIFLG) Field Descriptions**

Bit	Field	Value	Description
31-25	Reserved	0	Reads return 0. Writes have no effect.
24	BUFINITACTIVE	<p>0 Multi-buffer RAM initialization is complete.</p> <p>1 Multi-buffer RAM is still being initialized. Do not attempt to write to either multi-buffer RAM or any multi-buffer mode registers.</p>	<p>Indicates the status of multi-buffer initialization process. Software can poll for this bit to determine if it can proceed with the register configuration of multi-buffer mode registers or buffer handling.</p> <p><b>Note: If the SPIFLG register is read while the multi-buffer RAM is being initialized, the BUFINITACTIVE bit will be read as 1. If SPIFLG is read after the internal automatic buffer initialization is complete, this bit will be read as 0. This bit will show a value of 1 as long as the nRESET bit is 0, but does not really indicate that buffer initialization is underway. Buffer initialization starts only when the nRESET bit is set to 1.</b></p>
23-10	Reserved	0	Reads return 0. Writes have no effect.
9	TXINTFLG	<p>0 Transmit buffer is now full. No interrupt pending for transmitter empty.</p> <p>1 Transmit buffer is empty. An interrupt is pending to fill the transmitter.</p>	<p>Transmitter-empty interrupt flag. Serves as an interrupt flag indicating that the transmit buffer (TXBUF) is empty and a new word can be written to it. This flag is set when a word is copied to the shift register either directly from SPIDAT0/SPIDAT1 or from the TXBUF register. This bit is cleared by one of following methods:</p> <ul style="list-style-type: none"> <li>Writing a new data to either SPIDAT0 or SPIDAT1</li> <li>Writing a 0 to SPIEN (SPIGCR1[24])</li> </ul>

**Table 12-12. SPI Flag Register (SPIFLG) Field Descriptions (continued)**

Bit	Field	Value	Description
8	RXINTFLG	<div>0</div> <div>1</div>	<p>Receiver-full interrupt flag. This flag is set when a word is received and copied into the buffer register (SPIBUF). If RXINTEN is enabled, an interrupt is also generated. This bit is cleared under the following methods:</p> <ul style="list-style-type: none"><li>• Reading the SPIBUF register</li><li>• Reading TGINTVECT0 or TGINTVECT1 register when there is a receive buffer full interrupt</li><li>• Writing a 1 to this bit</li><li>• Writing a 0 to SPIEN (SPIGCR1[24])</li><li>• System reset</li></ul> <p>During emulation mode, however, a read to the emulation register (SPIEMU) does not clear this flag bit.</p> <p>No new received data pending. Receive buffer is empty.</p> <p>A newly received data is ready to be read. Receive buffer is full.</p> <p><b>Note: Clearing RXINTFLG bit by writing a 1 before reading the SPIBUF sets the RXEMPTY bit of the SPIBUF register too. In this way, one can ignore a received word. However, if the internal RXBUF is already full, the data from RXBUF will be copied to SPIBUF and the RXEMPTY bit will be cleared again. The SPIBUF contents should be read first if this situation needs to be avoided.</b></p>
7	Reserved	0	Reads return 0. Writes have no effect.
6	RXOVRNINTFLG	<div>0</div> <div>1</div>	<p>Receiver overrun flag. The SPI hardware sets this bit when a receive operation completes before the previous character has been read from the receive buffer. The bit indicates that the last received character has been overwritten and therefore lost. The SPI will generate an interrupt request if this bit is set and the RXOVRN INTEN bit (SPIINT0.6) is set high. This bit is cleared under the following conditions in compatibility mode of MibSPI:</p> <ul style="list-style-type: none"><li>• Reading TGINTVECT0 or TGINTVECT1 register when there is a receive-buffer-overrun interrupt</li><li>• Writing a 1 to RXOVRNINTFLG in the SPI Flag Register (SPIFLG) itself</li><li>• Writing a 0 to SPIEN</li><li>• Reading the data field of the SPIBUF register</li></ul> <p><b>Note: Reading the SPIBUF register does not clear this RXOVRNINTFLG bit. If an RXOVRN interrupt is detected, then the SPIBUF may need to be read twice to get to the overrun buffer. This is due to the fact that the overrun will always occur to the internal RXBUF. Each read to the SPIBUF will result in RXBUF contents (if it is full) getting copied to SPIBUF.</b></p> <p><b>Note: There is a special condition under which the RXOVRNINTFLG flag gets set. If both SPIBUF and RXBUF are already full and while another reception is underway, if any errors (TIMEOUT, BITERR, and DLEN_ERR) occur, then RXOVR in RXBUF and RXOVRNINTFLG in SPIFLG registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a receive overrun.</b></p> <p>In multi-buffer mode of MibSPI, this bit is cleared under the following conditions:</p> <ul style="list-style-type: none"><li>• Reading the RXOVRN_BUF_ADDR register</li><li>• Writing a 1 to RXOVRNINTFLG in the SPI Flag Register (SPIFLG) itself</li></ul> <p>In multi-buffer mode, if RXOVRNINTFLG is set, then the address of the buffer which experienced the overrun is available in RXOVRN_BUF_ADDR.</p> <p>Overrun condition did not occur.</p> <p>Overrun condition has occurred.</p>
5	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-12. SPI Flag Register (SPIFLG) Field Descriptions (continued)**

Bit	Field	Value	Description
4	BITERRFLG	<div>0</div> <div>1</div>	<p>Mismatch of internal transmit data and transmitted data. This flag can be cleared by one of the following methods:</p> <ul style="list-style-type: none"> <li>Write a 1 to this bit</li> <li>Clear the SPIEN bit to 0</li> </ul> <p>No bit error occurred.</p> <p>A bit error occurred. The SPI samples the signal of the transmit pin (master: SIMO, slave: SOMI) at the receive point (half clock cycle after transmit point). If the sampled value differs from the transmitted value a bit error is detected and the flag BITERRFLG is set. If BITERRFLG is set an interrupt is asserted. Possible reasons for a bit error can be an excessively high bit rate, capacitive load, or another master/slave trying to transmit at the same time.</p>
3	DESYNCFLG	<div>0</div> <div>1</div>	<p>Desynchronization of slave device. Desynchronization monitor is active in master mode only. This flag can be cleared by one of the following methods:</p> <ul style="list-style-type: none"> <li>Write a 1 to this bit</li> <li>Clear the SPIEN bit to 0</li> </ul> <p>No slave desynchronization is detected.</p> <p>A slave device is desynchronized. The master monitors the ENAble signal coming from the slave device and sets the DESYNC flag after the last bit is transmitted plus <math>t_{T2EDELAY}</math>. If DESYNCEA is set an interrupt is asserted. Desynchronization can occur if a slave device misses a clock edge coming from the master.</p>
2	PARERRFLG	<div>0</div> <div>1</div>	<p>Calculated parity differs from received parity bit. If the parity generator is enabled (can be selected individually for each buffer) an even or odd parity bit is added at the end of a data word. During reception of the data word the parity generator calculates the reference parity and compares it to the received parity bit. In the event of a mismatch the PARITYERR flag is set and an interrupt is asserted if PARERRFLG is set. This flag can be cleared by one of the following methods:</p> <ul style="list-style-type: none"> <li>Write a 1 to this bit</li> <li>Clear the SPIEN bit to 0</li> </ul> <p>No parity error is detected.</p> <p>A parity error occurred.</p>
1	TIMEOUTFLG	<div>0</div> <div>1</div>	<p>Time-out caused by nonactivation of ENA signal. This flag can be cleared by one of the following methods:</p> <ul style="list-style-type: none"> <li>Write a 1 to this bit</li> <li>Clear the SPIEN bit to 0</li> </ul> <p>No ENA-signal time-out occurred.</p> <p>An ENA signal time-out occurred. The SPI generates a time-out because the slave hasn't responded in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition the TIMEOUT flag in the status field of the corresponding buffer is set. The transmit request of the concerned buffer is cleared, that is, the SPI does not re-start a data transfer from this buffer.</p>
0	DLENERRFLG	<div>0</div> <div>1</div>	<p>Data-length error flag. This flag can be cleared by one of the following methods:</p> <ul style="list-style-type: none"> <li>Write a 1 to this bit</li> <li>Clear the SPIEN bit to 0</li> </ul> <p><b>Note: Whenever any transmission errors (TIMEOUT, BITERR, DLEN_ERR, PARITY_ERR, DESYNC) are detected and the error flags are cleared by writing to the error bit in the SPIFLG register, the corresponding error flag in SPIBUF does not get cleared. Software needs to read the SPIBUF until it becomes empty before proceeding. This ensures that all of the old status bits in SPIBUF are cleared before starting the next transfer.</b></p> <p>No data length error has occurred.</p> <p>A data length error has occurred.</p>

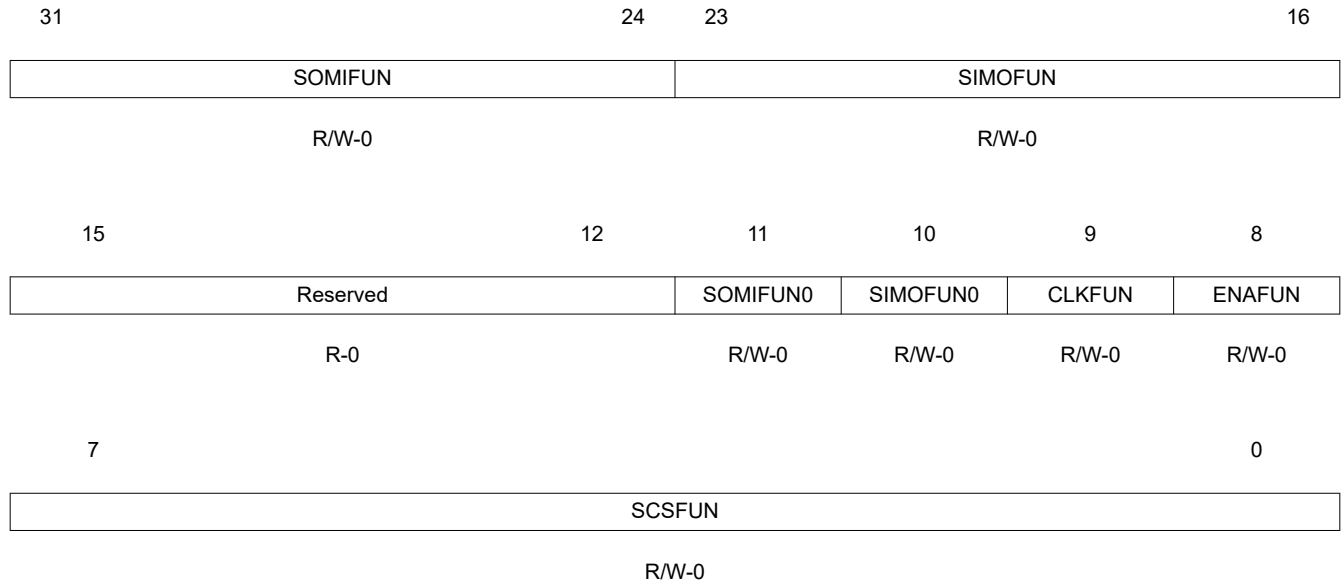
## 12.8.6 SPI Pin Control Register 0 (SPIPC0)

### Note

#### Register bits vary by device

Register bits 31:24 and 23:16 of SPIPC0 to SPIPC9 reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-31. SPI Pin Control Register 0 (SPIPC0) [offset = 14h]**



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-13. SPI Pin Control (SPIPC0) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMIFUN	<div> <div></div> <div>0</div> <div>1</div> </div>	<p>Slave out, master in function. Determines whether each SPISOMI[x] pin is to be used as a general-purpose I/O pin or as a SPI functional pin.</p> <p><b>Note: Duplicate Control Bits for SPISOMI[0]. Bit 24 is not physically implemented. It is a mirror of Bit 11. Any write to bit 24 will be reflected on bit 11. When bit 24 and bit 11 are simultaneously written, the value of bit 11 will control the SPISOMI[0] pin. The read value of bit 24 always reflects the value of bit 11.</b></p> <p>The SPISOMI[x] pin is a GIO pin.</p> <p>The SPISOMI[x] pin is a SPI functional pin.</p>
23-16	SIMOFUN	<div> <div></div> <div>0</div> <div>1</div> </div>	<p>Slave in, master out function. Determines whether each SPISIMO[x] pin is to be used as a general-purpose I/O pin or as a SPI functional pin.</p> <p><b>Note: Duplicate Control Bits for SPISIMO[0]. Bit 16 is not physically implemented. It is a mirror of Bit 10. Any write to bit 16 will be reflected on bit 10. When bit 16 and bit 10 are simultaneously written, the value of bit 10 will control the SPISIMO[0] pin. The read value of bit 16 always reflects the value of bit 10.</b></p> <p>The SPISIMO[x] pin is a GIO pin.</p> <p>The SPISIMO[x] pin is a SPI functional pin.</p>
15-12	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-13. SPI Pin Control (SPIPC0) Field Descriptions (continued)**

Bit	Field	Value	Description
11	SOMIFUN0	0 1	Slave out, master in function. This bit determines whether the SPISOMI[0] pin is to be used as a general-purpose I/O pin or as a SPI functional pin. The SPISOMI[0] pin is a GIO pin. The SPISOMI[0] pin is a SPI functional pin. <b>Note: Regardless of the number of parallel pins used, the SPISOMI[0] pin will always have to be programmed as functional pins for any SPI transfers.</b>
10	SIMOFUN0	0 1	Slave in, master out function. This bits determine whether each SPISIMO[0] pin is to be used as a general-purpose I/O pin or as a SPI functional pin. The SPISIMO[0] pin is a GIO pin. The SPISIMO[0] pin is a SPI functional pin. <b>Note: Regardless of the number of parallel pins used, the SPISIMO[0] pin will always have to be programmed as functional pins for any SPI transfers.</b>
9	CLKFUN	0 1	SPI clock function. This bit determines whether the SPICLK pin is to be used as a general-purpose I/O pin, or as a SPI functional pin. The SPICLK pin is a GIO pin. The SPICLK pin is a SPI functional pin.
8	ENAFUN	0 1	SPIEN $\overline{A}$ function. This bit determines whether the SPIEN $\overline{A}$ pin is to be used as a general-purpose I/O pin or as a SPI functional pin. The SPIEN $\overline{A}$ pin is a GIO pin. The SPIEN $\overline{A}$ pin is a SPI functional pin.
7-0	SCSFUN	0 1	SPICS function. Determines whether each SPICS pin is to be used as a general-purpose I/O pin or as a SPI functional pin. If the slave SPICS pins are in functional mode and receive an inactive high signal, the slave SPI will place its output in a high-impedance state and disable shifting. The SPICS pin is a GIO pin. The SPICS pin is a SPI functional pin.

### 12.8.7 SPI Pin Control Register 1 (SPIPC1)

#### Note

#### Register bits vary by device

Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-32. SPI Pin Control Register 1 (SPIPC1) [offset = 18h]**

31	24	23			16
SOMIDIR			SIMODIR		
R/W-0			R/W-0		
15	12	11	10	9	8
Reserved		SOMIDIR0	SIMODIR0	CLKDIR	ENADIR
R-0		R/W-0	R/W-0	R/W-0	R/W-0

7

0

SCSDIR

R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-14. SPI Pin Control Register (SPIPC1) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMIDIR	<p>0 The SPISOMI[x] pin is an input.</p> <p>1 The SPISOMI[x] pin is an output.</p>	<p>SPISOMI[x] direction. Controls the direction of each SPISOMI[x] pin when used for general-purpose I/O. If SPISOMI[x] pin is used as a SPI functional pin, the I/O direction is determined by the MASTER bit in the SPIGCR1 register.</p> <p><b>Note: Duplicate Control Bits for SPISOMI[0]. Bit 24 is not physically implemented. It is a mirror of Bit 11. Any write to bit 24 will be reflected on bit 11. When bit 24 and bit 11 are simultaneously written, the value of bit 11 will control the SPISOMI[0] pin. The read value of bit 24 always reflects the value of bit 11.</b></p>
23-16	SIMODIR	<p>0 The SPISIMO[x] pin is an input.</p> <p>1 The SPISIMO[x] pin is an output.</p>	<p>SPISIMO[x] direction. Controls the direction of each SPISIMO[x] pin when used for general-purpose I/O. If SPISIMO[x] pin is used as a SPI functional pin, the I/O direction is determined by the MASTER bit in the SPIGCR1 register.</p> <p><b>Note: Duplicate Control Bits for SPISIMO[0]. Bit 16 is not physically implemented. It is a mirror of Bit 10. Any write to bit 16 will be reflected on bit 10. When bit 16 and bit 10 are simultaneously written, the value of bit 10 will control the SPISIMO[0] pin. The read value of bit 16 always reflects the value of bit 10.</b></p>
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SOMIDIR0	<p>0 The SPISOMI[0] pin is an input.</p> <p>1 The SPISOMI[0] pin is an output.</p>	<p>SPISOMI[0] direction. This bit controls the direction of the SPISOMI[0] pin when it is used as a general-purpose I/O pin. If the SPISOMI[0] pin is used as a SPI functional pin, the I/O direction is determined by the MASTER bit in the SPIGCR1 register.</p>
10	SIMODIR0	<p>0 The SPISIMO[0] pin is an input.</p> <p>1 The SPISIMO[0] pin is an output.</p>	<p>SPISIMO[0] direction. This bit controls the direction of the SPISIMO[0] pin when it is used as a general-purpose I/O pin. If the SPISIMO[0] pin is used as a SPI functional pin, the I/O direction is determined by the MASTER bit in the SPIGCR1 register.</p>
9	CLKDIR	<p>0 The SPICLK pin is an input.</p> <p>1 The SPICLK pin is an output.</p>	<p>SPICLK direction. This bit controls the direction of the SPICLK pin when it is used as a general-purpose I/O pin. In functional mode, the I/O direction is determined by the CLKMOD bit.</p>
8	ENADIR	<p>0 The <math>\overline{\text{SPIEN}}\text{A}</math> pin is an input.</p> <p>1 The <math>\overline{\text{SPIEN}}\text{A}</math> pin is an output.</p>	<p><math>\overline{\text{SPIEN}}\text{A}</math> direction. This bit controls the direction of the <math>\overline{\text{SPIEN}}\text{A}</math> pin when it is used as a general-purpose I/O. If the <math>\overline{\text{SPIEN}}\text{A}</math> pin is used as a functional pin, then the I/O direction is determined by the CLKMOD bit (SPIGCR1[1]).</p>
7-0	SCSDIR	<p>0 The <math>\overline{\text{SPICS}}</math> pin is an input.</p> <p>1 The <math>\overline{\text{SPICS}}</math> pin is an output.</p>	<p><math>\overline{\text{SPICS}}</math> direction. These bits control the direction of each <math>\overline{\text{SPICS}}</math> pin when it is used as a general-purpose I/O pin. Each pin could be configured independently from the others if the <math>\overline{\text{SPICS}}</math> is used as a SPI functional pin. The I/O direction is determined by the CLKMOD bit (SPIGCR1[1]).</p>

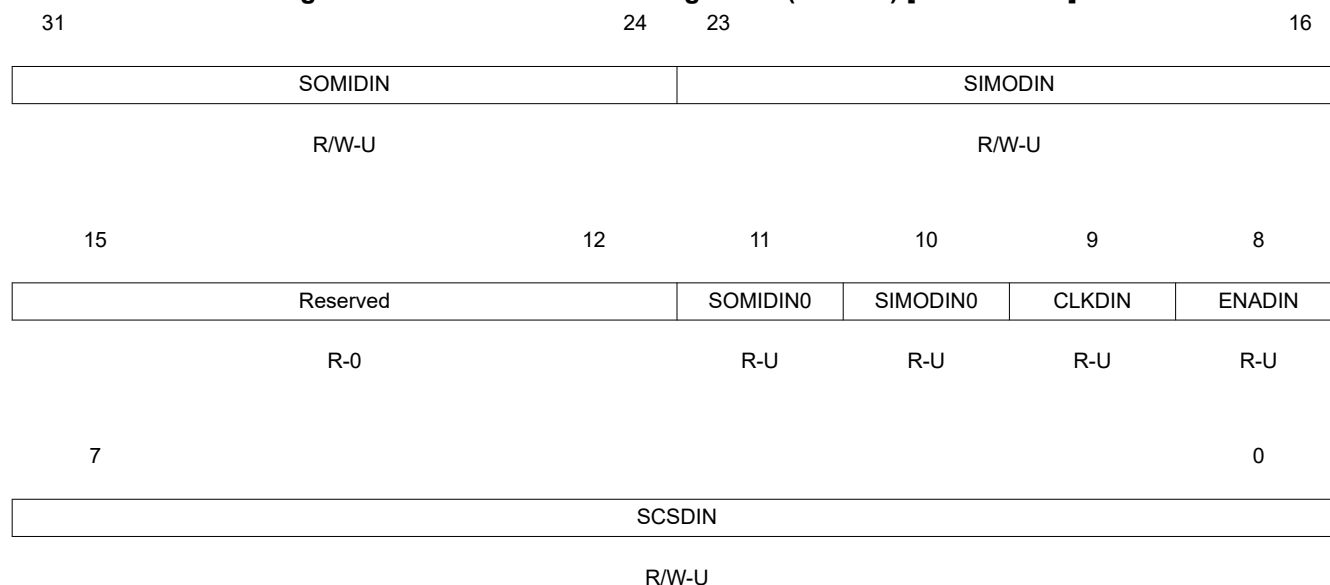
## 12.8.8 SPI Pin Control Register 2 (SPIPC2)

### Note

#### Register bits vary by device

Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-33. SPI Pin Control Register 2 (SPIPC2) [offset = 1Ch]**



LEGEND: R/W = Read/Write; R = Read only; U = Undefined; -n = value after reset

**Table 12-15. SPI Pin Control Register 2 (SPIPC2) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMIDIN	0 1	SPISOMI[x] data in. The value of each SPISOMI[x] pin. The SPISOMI[x] pin is logic 0. The SPISOMI[x] pin is logic 1.
23-16	SIMODIN	0 1	SPISIMO[x] data in. The value of each SPISIMO[x] pin. The SPISIMO[x] pin is logic 0. The SPISIMO[x] pin is logic 1.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SOMIDIN0	0 1	SPISOMI[0] data in. The value of the SPISOMI[0] pin. The SPISOMI[0] pin is logic 0. The SPISOMI[0] pin is logic 1.
10	SIMODIN0	0 1	SPISIMO[0] data in. The value of the SPISIMO[0] pin. The SPISIMO[0] pin is logic 0. The SPISIMO[0] pin is logic 1.
9	CLKDIN	0 1	Clock data in. The value of the SPICLK pin. The SPICLK pin is logic 0. The SPICLK pin is logic 1.



**Table 12-15. SPI Pin Control Register 2 (SPIPC2) Field Descriptions (continued)**

Bit	Field	Value	Description
8	ENADIN		SPIEN $\overline{A}$ data in. The the value of the SPIEN $\overline{A}$ pin.
		0	The SPIEN $\overline{A}$ pin is logic 0.
		1	The SPIEN $\overline{A}$ pin is logic 1.
7-0	SCSDIN		SPICS data in. The value of each SPICS pin.
		0	The SPICS pin is logic 0.
		1	The SPICS pin is logic 1.

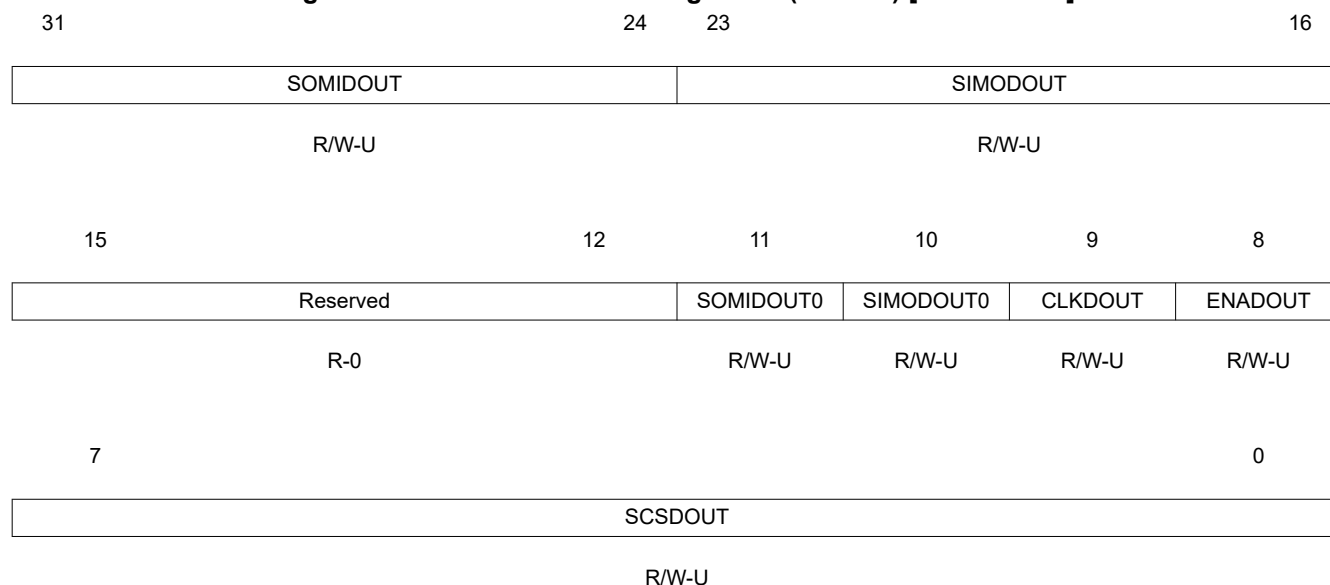
### 12.8.9 SPI Pin Control Register 3 (SPIPC3)

#### Note

#### Register bits vary by device

Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-34. SPI Pin Control Register 3 (SPIPC3) [offset = 20h]**



LEGEND: R/W = Read/Write; R = Read only; U = Undefined; -n = value after reset

**Table 12-16. SPI Pin Control Register 3 (SPIPC3) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMIDOUT	0 Current value on SPISOMI[x] pin is logic 0. 1 Current value on SPISOMI[x] pin is logic 1	SPISOMI[x] data out write. This bit is only active when the SPISOMI[x] pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin.  <b>Bit 11 or bit 24 can be used to set the direction for pin SPISOMI[0]. If a 32-bit write is performed, bit 11 will have priority over bit 24.</b>
23-16	SIMODOUT	0 Current value on SPISIMO[x] pin is logic 0. 1 Current value on SPISIMO[x] pin is logic 1.	SPISIMO[x] data out write. This bit is only active when the SPISIMO[x] pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin.  <b>Bit 10 or bit 16 can be used to set the direction for pin SPISIMO[0]. If a 32-bit write is performed, bit 10 will have priority over bit 16.</b>
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SOMIDOUT0	0 Current value on SPISOMI[0] pin is logic 0. 1 Current value on SPISOMI[0] pin is logic 1.	SPISOMI[0] data out write. This bit is only active when the SPISOMI[0] pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin.

**Table 12-16. SPI Pin Control Register 3 (SPIPC3) Field Descriptions (continued)**

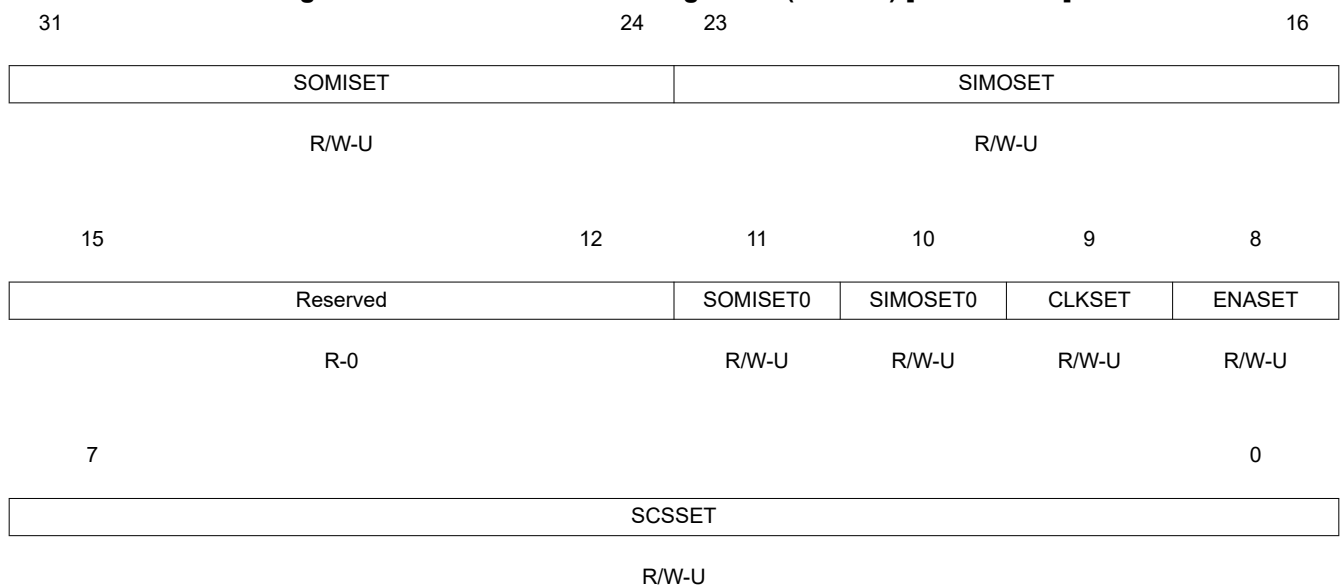
Bit	Field	Value	Description
10	SIMODOUT0	0 1	SPISIMO[0] data out write. This bit is only active when the SPISIMO[0] pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. Current value on SPISIMO[0] pin is logic 0. Current value on SPISIMO[0] pin is logic 1.
9	CLKDOUT	0 1	SPICLK data out write. This bit is only active when the SPICLK pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. The SPICLK pin is logic 0. The SPICLK pin is logic 1.
8	ENADOUT	0 1	SPIEN $\overline{A}$ data out write. Only active when the SPIEN $\overline{A}$ pin is configured as a general-purpose I/O pin and configured as an output pin. The value of this bit indicates the value sent to the pin. The SPIEN $\overline{A}$ pin is logic 0. The SPIEN $\overline{A}$ pin is logic 1.
7-0	SCSDOUT	0 1	SPIC $\overline{S}$ data out write. Only active when the SPIC $\overline{S}$ pins are configured as a general-purpose I/O pins and configured as output pins. The value of these bits indicates the value sent to the pins. The SPIC $\overline{S}$ pin is logic 0. The SPIC $\overline{S}$ pin is logic 1.

#### 12.8.10 SPI Pin Control Register 4 (SPIPC4)

##### Note

##### Register bits vary by device

Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-35. SPI Pin Control Register 4 (SPIPC4) [offset = 24h]**


LEGEND: R/W = Read/Write; R = Read only; U = Undefined; -n = value after reset

**Table 12-17. SPI Pin Control Register 4 (SPIPC4) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMISET	0	SPISOMI[x] data out set. This pin is only active when the SPISOMI[x] pin is configured as a general-purpose output pin. <b>Bit 11 or bit 24 can be used to set the SPISOMI[0] pin. If a 32-bit write is performed, bit 11 will have priority over bit 24.</b> Read: SPISOMI[x] is logic 0. Write: Writing a 0 to this bit has no effect.
		1	Read: SPISOMI[x] is logic 1. Write: Logic 1 is placed on SPISOMI[x] pin, if it is in general-purpose output mode.
23-16	SIMOSET	0	SPISIMO[x] data out set. This bit is only active when the SPISIMO[x] pin is configured as a general-purpose output pin. <b>Bit 10 or bit 16 can be used to set the SPISIMO[0] pin. If a 32-bit write is performed, bit 10 will have priority over bit 16.</b> Read: SPISIMO[x] is logic 0. Write: Writing a 0 to this bit has no effect.
		1	Read: SPISIMO[x] is logic 1. Write: Logic 1 is placed on SPISIMO[x] pin, if it is in general-purpose output mode.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SOMISET0	0	SPISOMI[0] data out set. This pin is only active when the SPISOMI[0] pin is configured as a general-purpose output pin. Read: SPISOMI[0] is logic 0. Write: Writing a 0 to this bit has no effect.
		1	Read: SPISOMI[0] is logic 1. Write: Logic 1 is placed on SPISOMI[0] pin, if it is in general-purpose output mode.
10	SIMOSET0	0	SPISIMO[0] data out set. This pin is only active when the SPISIMO[0] pin is configured as a general-purpose output pin. Read: SPISIMO[0] is logic 0. Write: Writing a 0 to this bit has no effect.
		1	Read: SPISIMO[0] is logic 1. Write: Logic 1 is placed on SPISIMO[0] pin, if it is in general-purpose output mode.
9	CLKSET	0	SPICLK data out set. This bit is only active when the SPICLK pin is configured as a general-purpose output pin. Read: SPICLK is logic 0. Write: Writing a 0 to this bit has no effect.
		1	Read: SPICLK is logic 1. Write: Logic 1 is placed on SPICLK pin, if it is in general-purpose output mode.
8	ENASET	0	$\overline{\text{SPIEN}}\text{A}$ data out set. This bit is only active when the $\overline{\text{SPIEN}}\text{A}$ pin is configured as a general-purpose output pin. Read: $\overline{\text{SPIEN}}\text{A}$ is logic 0. Write: Writing a 0 to this bit has no effect.
		1	Read: $\overline{\text{SPIEN}}\text{A}$ is logic 1. Write: Logic 1 is placed on $\overline{\text{SPIEN}}\text{A}$ pin, if it is in general-purpose output mode.

**Table 12-17. SPI Pin Control Register 4 (SPIPC4) Field Descriptions (continued)**

Bit	Field	Value	Description
7-0	SCSSET	0	<p><math>\overline{\text{SPICS}}</math> data out set. This bit is only active when the <math>\overline{\text{SPICS}}</math> pin is configured as a general-purpose output pin. A value of 1 written to this bit sets the corresponding SCSDOUT bit to 1.</p> <p>Read: <math>\overline{\text{SPICS}}</math> is logic 0.</p> <p>Write: Writing a 0 to this bit has no effect.</p>
		1	<p>Read: <math>\overline{\text{SPICS}}</math> is logic 1.</p> <p>Write: Logic 1 is placed on <math>\overline{\text{SPICS}}</math> pin, if it is in general-purpose output mode.</p>

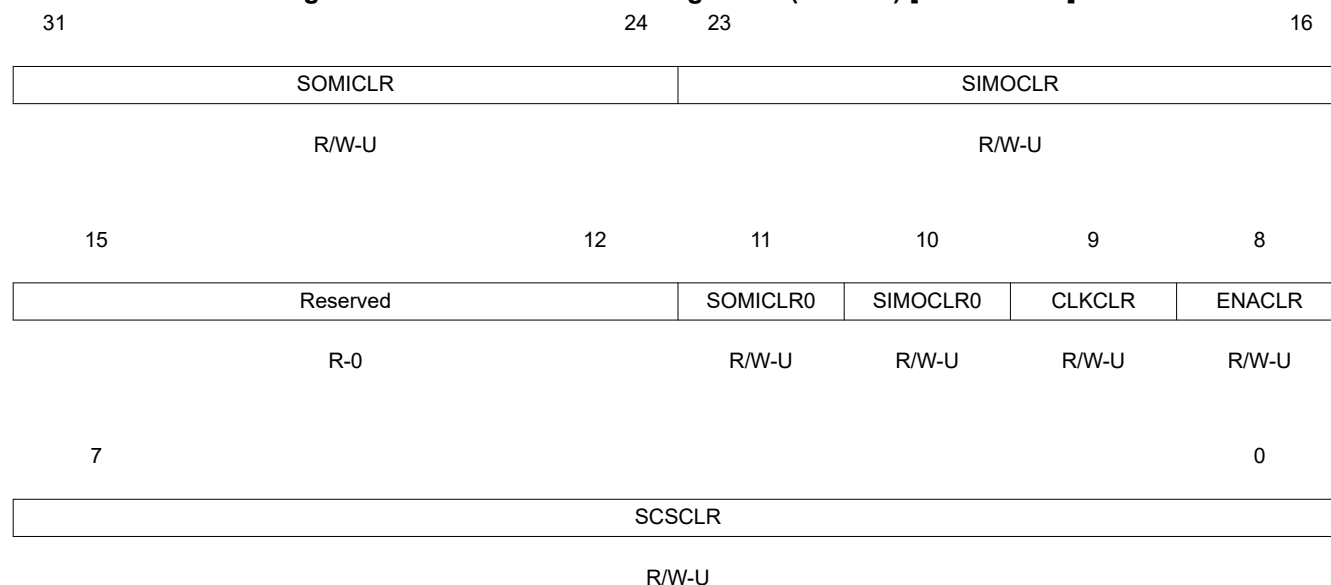
## 12.8.11 SPI Pin Control Register 5 (SPIPC5)

### Note

#### Register bits vary by device

Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-36. SPI Pin Control Register 5 (SPIPC5) [offset = 28h]**



LEGEND: R/W = Read/Write; R = Read only; U = Undefined; -n = value after reset

**Table 12-18. SPI Pin Control Register 5 (SPIPC5) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMICLR		SPISOMI[x] data out clear. This pin is only active when the SPISOMI[x] pin is configured as a general-purpose output pin. <b>Bit 11 or bit 24 can be used to set the SPISOMI[0] pin. If a 32-bit write is performed, bit 11 will have priority over bit 24.</b>
		0	Read: The current value on SPISOMI[x] is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SPISOMI[x] is 1. Write: Logic 0 is placed on SPISOMI[x] pin, if it is in general-purpose output mode.
23-16	SIMOCLR		SPISIMO[x] data out clear. This bit is only active when the SPISIMO[x] pin is configured as a general-purpose output pin. <b>Bit 10 or bit 16 can be used to set the SPISIMO[0] pin. If a 32-bit write is performed, bit 10 will have priority over bit 16.</b>
		0	Read: The current value on SPISIMO[x] is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SPISIMO[x] is 1. Write: Logic 0 is placed on SPISIMO[x] pin, if it is in general-purpose output mode.
15-12	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-18. SPI Pin Control Register 5 (SPIPC5) Field Descriptions (continued)**

Bit	Field	Value	Description
11	SOMICLR0	0	SPISOMI[0] data out clear. This pin is only active when the SPISOMI[0] pin is configured as a general-purpose output pin. Read: The current value on SPISOMI[0] is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SPISOMI[0] is 1. Write: Logic 0 is placed on SPISOMI[0] pin, if it is in general-purpose output mode.
10	SIMOCLR0	0	SPISIMO[0] data out clear. This pin is only active when the SPISIMO[0] pin is configured as a general-purpose output pin. Read: The current value on SPISIMO[0] is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SPISIMO[0] is 1. Write: Logic 0 is placed on SPISIMO[0] pin, if it is in general-purpose output mode.
9	CLKCLR	0	SPICLK data out clear. This bit is only active when the SPICLK pin is configured as a general-purpose output pin. Read: The current value on SPICLK is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SPICLK is 1. Write: Logic 0 is placed on SPICLK pin, if it is in general-purpose output mode.
8	ENACLR	0	SPIEN $\overline{A}$ data out clear. This bit is only active when the SPIEN $\overline{A}$ pin is configured as a general-purpose output pin. A value of 1 written to this bit clears the corresponding ENABLEDOUT bit to 0. Read: The current value on SPIEN $\overline{A}$ is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SPIEN $\overline{A}$ is 1. Write: Logic 0 is placed on SPIEN $\overline{A}$ pin, if it is in general-purpose output mode.
7-0	SCSCLR	0	SPICS data out clear. This bit is only active when the SPICS pin is configured as a general-purpose output pin. Read: The current value on SCSDOUT is 0. Write: Writing a 0 to this bit has no effect.
		1	Read: The current value on SCSDOUT is 1. Write: Logic 0 is placed on SPICS pin, if it is in general-purpose output mode.

### 12.8.12 SPI Pin Control Register 6 (SPIPC6)

#### Note

#### Register bits vary by device

Register bits 31:24 and 23:16 of SPIPC0 to SPIPC9 reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

**Figure 12-37. SPI Pin Control Register 6 (SPIPC6) [offset = 2Ch]**

31	24	23	16
SOMIPDR		SIMOPDR	
R/W-0		R/W-0	

15	12	11	10	9	8
Reserved		SOMIPDR0	SIMOPDR0	CLKPDR	ENAPDR
R-0		R/W-0	R/W-0	R/W-0	R/W-0
7					0
SCSPDR					
R/W-0					

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-19. SPI Pin Control Register 6 (SPIPC6) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMIPDR	0 1	<p>SPISOMI[x] open drain enable. This bit enables open drain capability for each SPISOMI[x] pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>SOMIDIRx = 1 (SPISOMI[x] pin is configured in GIO mode as an output pin)</li> <li>SOMIDOUTx = 1</li> </ul> <p><b>Bit 11 or bit 24 can both be used to enable open-drain for SPISOMI[0]. If a 32-bit write is performed, bit 11 will have priority over bit 24.</b></p> <p>Output value on the SPISOMI[x] pin is logic 1.</p> <p>Output pin SPISOMI[x] is in a high-impedance state.</p>
23-16	SIMOPDR	0 1	<p>SPISIMO[x] open drain enable. This bit enables open drain capability for each SPISIMO[x] pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>SIMODIRx = 1 (SPISIMO[x] pin is configured in GIO mode as an output pin)</li> <li>SIMODOUTx = 1</li> </ul> <p><b>Bit 10 or bit 16 can both be used to enable open-drain for SPISIMO[0]. If a 32-bit write is performed, bit 10 will have priority over bit 16.</b></p> <p>Output value on the SPISIMO[x] pin is logic 1.</p> <p>Output pin SPISIMO[x] is in a high-impedance state.</p>
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SOMIPDR0	0 1	<p>SPISOMI[0] open-drain enable. This bit enables open-drain capability for the SPISOMI[0] pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>SPISOMI[0] pin is configured in GIO mode as output pin</li> <li>Output value on SPISOMI[0] pin is logic 1</li> </ul> <p>Output value on the SPISOMI[0] pin is logic 1.</p> <p>Output pin SPISOMI[0] is in a high-impedance state.</p>
10	SIMOPDR0	0 1	<p>SPISIMO[0] open-drain enable. This bit enables open drain capability for the SPISIMO[0] pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>SPISIMO[0] pin is configured in GIO mode as output pin</li> <li>Output value on SPISIMO[0] pin is logic 1</li> </ul> <p>Output value on the SPISIMO[0] pin is logic 1.</p> <p>Output pin SPISIMO[0] is in a high-impedance state.</p>



**Table 12-19. SPI Pin Control Register 6 (SPIPC6) Field Descriptions (continued)**

Bit	Field	Value	Description
9	CLKPDR	<div> <div></div> <div>0</div> <div>1</div> </div>	<p>SPICLK open drain enable. This bit enables open drain capability for the SPICLK pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>• SPICLK pin is configured in GIO mode as an output pin</li> <li>• SPICLKDOOUT = 1</li> </ul> <p>Output value on the SPICLK pin is logic 1.</p> <p>Output pin SPICLK is in a high-impedance state.</p>
8	ENAPDR	<div> <div></div> <div>0</div> <div>1</div> </div>	<p><math>\overline{\text{SPIEN}}\overline{\text{A}}</math> open drain enable. This bit enables open drain capability for the <math>\overline{\text{SPIEN}}\overline{\text{A}}</math> pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>• <math>\overline{\text{SPIEN}}\overline{\text{A}}</math> pin is configured in GIO mode as an output pin</li> <li>• SPIENADOUT = 1</li> </ul> <p>Output value on the <math>\overline{\text{SPIEN}}\overline{\text{A}}</math> pin is logic 1.</p> <p>Output pin <math>\overline{\text{SPIEN}}\overline{\text{A}}</math> is in a high-impedance state.</p>
7-0	SCSPDR	<div> <div></div> <div>0</div> <div>1</div> </div>	<p><math>\overline{\text{SPICS}}</math> open drain enable. This bit enables open drain capability for each <math>\overline{\text{SPICS}}</math> pin, if the following conditions are met:</p> <ul style="list-style-type: none"> <li>• <math>\overline{\text{SPICS}}</math> pin is configured in GIO mode as an output pin</li> <li>• SCSDOUT = 1</li> </ul> <p>Output value on the <math>\overline{\text{SPICS}}</math> pin is logic 1.</p> <p>Output pin <math>\overline{\text{SPICS}}</math> is in a high-impedance state.</p>

### 12.8.13 SPI Pin Control Register 7 (SPIPC7)

#### Note

#### Register bits vary by device

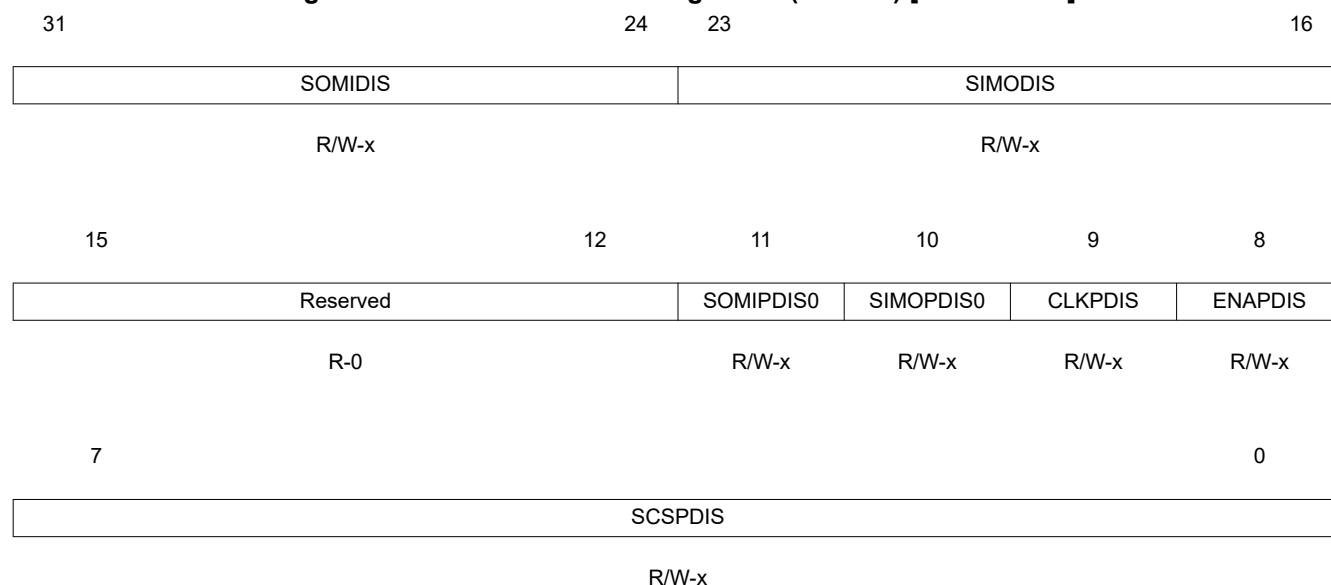
Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

#### Note

#### Default Register Value

The default values of these register bits vary by device. See your device datasheet for information about default pin states, which correspond to the register reset values (see the pin-list table).

**Figure 12-38. SPI Pin Control Register 7 (SPIPC7) [offset = 30h]**



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset; -x = value varies by device

**Table 12-20. SPI Pin Control Register 7 (SPIPC7) Field Descriptions**

Bit	Field	Value	Description
31-24	SOMIDIS	<div>0</div> <div>1</div>	<div>SPISOMI[x] pull control disable. This bit disables pull control capability for each SPISOMI[x] pin if it is in input mode, regardless of whether it is in functional or GIO mode.</div> <div><b>Note: Bit 11 or bit 24 can be used to set pull-disable for SPISOMI[0]. If a 32-bit write is performed, bit 11 will have priority over bit 24.</b></div> <div>Pull control on the SPISOMI[x] pin is enabled.</div> <div>Pull control on the SPISOMI[x] pin is disabled.</div>
23-16	SIMODIS	<div>0</div> <div>1</div>	<div>SPISIMO[x] pull control disable. This bit disables pull control capability for each SPISIMO[x] pin if it is in input mode, regardless of whether it is in functional or GIO mode.</div> <div><b>Note: Bit 10 or bit 16 can be used to set pull-disable for SPISIMO[0]. If a 32-bit write is performed, bit 10 will have priority over bit 16.</b></div> <div>Pull control on the SPISIMO[x] pin is enabled.</div> <div>Pull control on the SPISIMO[x] pin is disabled.</div>
15-12	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-20. SPI Pin Control Register 7 (SPIPC7) Field Descriptions (continued)**

Bit	Field	Value	Description
11	SOMIPDIS0	0 1	SPISOMI[0] pull control disable. This bit disables pull control capability for the SPISOMI[0] pin if it is in input mode, regardless of whether it is in functional or GIO mode. Pull control on the SPISOMI[0] pin is enabled. Pull control on the SPISOMI[0] pin is disabled.
10	SIMOPDIS0	0 1	SPISIMO[0] pull control disable. This bit disables pull control capability for the SPISIMO[0] pin if it is in input mode, regardless of whether it is in functional or GIO mode. Pull control on the SPISIMO[0] pin is enabled. Pull control on the SPISIMO[0] pin is disabled.
9	CLKPDIS	0 1	SPICLK pull control disable. This bit disables pull control capability for the SPICLK pin if it is in input mode, regardless of whether it is in functional or GIO mode. Pull control on the SPICLK pin is enabled. Pull control on the SPICLK pin is disabled.
8	ENAPDIS	0 1	SPIEN $\bar{A}$ pull control disable. This bit disables pull control capability for the SPIEN $\bar{A}$ pin if it is in input mode, regardless of whether it is in functional or GIO mode. Pull control on the SPIEN $\bar{A}$ pin is enabled. Pull control on the SPIEN $\bar{A}$ pin is disabled.
7-0	SCSPDIS	0 1	SPICS pull control disable. This bit disables pull control capability for each SPICS pin if it is in input mode, regardless of whether it is in functional or GIO mode. Pull control on the SPICS pin is enabled. Pull control on the SPICS pin is disabled.

#### 12.8.14 SPI Pin Control Register 8 (SPIPC8)

##### Note

##### Register bits vary by device

Register bits 31:24 and 23:16 of this register reflect the number of SIMO/SOMI data lines per device. On devices with 8 data-line support, all of bits 31 to 16 are implemented. On devices with less than 8 data lines, only a subset of these bits are available. Unimplemented bits return 0 upon read and are not writable.

##### Note

##### Default Register Value

The default values of these register bits vary by device. See your device datasheet for information about default pin states, which correspond to the register reset values (see the pin-list table).

**Figure 12-39. SPI Pin Control Register 8 (SPIPC8) [offset = 34h]**

31		24	23					16
SOMIPSEL				SIMOPSEL				
R/W-x				R/W-x				
15		12	11	10	9			8
Reserved				SOMIPSEL0	SIMOPSEL0	CLKPSEL	ENAPSEL	

R/W-x

0

R/W-x

15

0

TXDATA

R/W-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-22. SPI Transmit Data Register 0 (SPIDAT0) Field Descriptions**

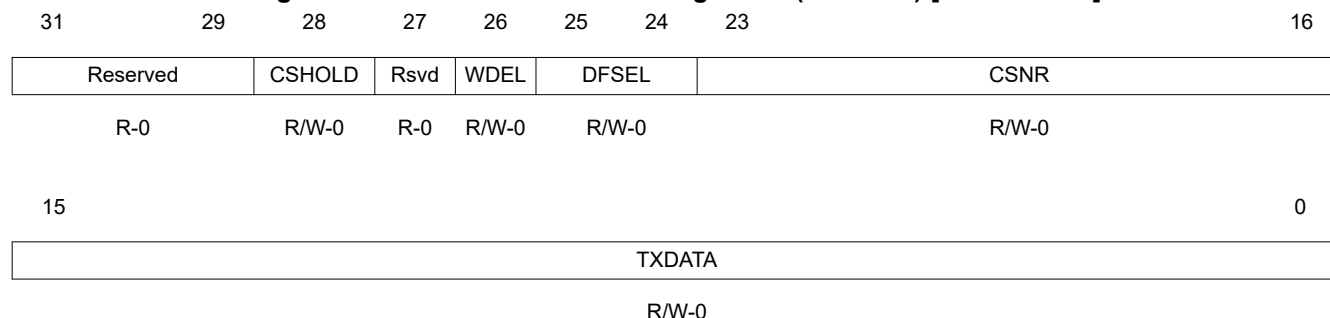
Bit	Field	Value	Description
31-16	Reserved	0	Reads return 0. Writes have no effect.
15-0	TXDATA	0-FFFFh	<p>SPI transmit data. When written, these bits will be copied to the shift register if it is empty. If the shift register is not empty, TXBUF holds the written data. SPIEN (SPICGR1[24]) must be set to 1 before this register can be written to. Writing a 0 to the SPIEN register forces the lower 16 bits of the SPIDAT0 to 0x00.</p> <p><b>Note:</b> When this register is read, the contents TXBUF, which holds the latest written data, will be returned.</p> <p><b>Note:</b> Regardless of character length, the transmit word should be right-justified before writing to the SPIDAT1 register.</p> <p><b>Note:</b> The default data format control register for SPIDAT0 is SPIFMT0. However, it is possible to reprogram the DFSEL[1:0] fields of SPIDAT1 before using SPIDAT0, to select a different SPIFMTx register.</p> <p><b>Note:</b> It is highly recommended to use SPIDAT1 register, SPIDAT0 is supported for compatibility reasons.</p>

### 12.8.16 SPI Transmit Data Register 1 (SPIDAT1)

#### Note

Writing to only the control fields, bits 28 through 16, does not initiate any SPI transfer in master mode. This feature can be used to set up SPICLK phase or polarity before actually starting the transfer by only updating the DFSEL bit field to select the required phase and polarity combination.

**Figure 12-41. SPI Transmit Data Register 1 (SPIDAT1) [offset = 3Ch]**



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-23. SPI Transmit Data Register 1 (SPIDAT1) Field Descriptions**

Bit	Field	Value	Description
31-29	Reserved	0	Reads return 0. Writes have no effect.
28	CSHOLD	0	Chip select hold mode. The CSHOLD bit is supported in master mode only in compatibility-mode of SPI, (it is ignored in slave mode). CSHOLD defines the behavior of the chip select line at the end of a data transfer.  The chip select signal is deactivated at the end of a transfer after the T2CDELAY time has passed. If two consecutive transfers are dedicated to the same chip select this chip select signal will be deactivated for at least 2VCLK cycles before it is activated again.
		1	The chip select signal is held active at the end of a transfer until a control field with new data and control information is loaded into SPIDAT1. If the new chip select number equals the previous one, the active chip select signal is extended until the end of transfer with CSHOLD cleared, or until the chip-select number changes.
27	Reserved	0	Reads return 0. Writes have no effect.
26	WDEL	0	Enable the delay counter at the end of the current transaction.  <b>Note: The WDEL bit is supported in master mode only. In slave mode, this bit will be ignored.</b>  No delay will be inserted. However, the $\overline{\text{SPICS}}$ pins will still be de-activated for at least for 2VCLK cycles if CSHOLD = 0.
		1	<b>Note: The duration for which the <math>\overline{\text{SPICS}}</math> pin remains deactivated depends upon the time taken to supply a new word after completing the shift operation. If TXBUF is already full, then the <math>\overline{\text{SPICS}}</math> pin will be deasserted for at least two VCLK cycles (if WDEL = 0).</b>  After a transaction, WDELAY of the corresponding data format will be loaded into the delay counter. No transaction will be performed until the WDELAY counter overflows. The $\overline{\text{SPICS}}$ pins will be de-activated for at least (WDELAY + 2) × VCLK_Period duration.
25-24	DFSEL	0	Data word format 0 is selected.
		1h	Data word format 1 is selected.
		2h	Data word format 2 is selected.
		3h	Data word format 3 is selected.

**Table 12-23. SPI Transmit Data Register 1 (SPIDAT1) Field Descriptions (continued)**

Bit	Field	Value	Description
23-16	CSNR	0-FFh	<p>Chip select (CS) number. CSNR defines the chip select pins that will be activated during the data transfer. CSNR is a bit-mask that controls all chip select pins. See <a href="#">Table 12-24</a>.</p> <p><b>Note: If your MibSPI has less than 8 chip select pins, all unused upper bits will be 0. For example, MiBSPi3 has 6 chip select pins, if you write FFh to CSNR, the actual number stored in CSNR is 3Fh.</b></p>
15-0	TXDATA	0-FFFFh	<p>Transfer data. When written, these bits are copied to the shift register if it is empty. If the shift register is not empty, then they are held in TXBUF.</p> <p>SPIEN must be set to 1 before this register can be written to. Writing a 0 to SPIEN forces the lower 16 bits of SPIDAT1 to 0x0000.</p> <p>A write to this register (or to the TXDATA field only) drives the contents of the CSNR field on the SPICS pins, if the pins are configured as functional pins (automatic chip select, see <a href="#">Section 12.2</a>).</p> <p>When this register is read, the contents of TXBUF, which holds the latest data written, will be returned.</p> <p><b>Note: Regardless of the character length, the transmit data should be right-justified before writing to the SPIDAT1 register.</b></p>

**Table 12-24. Chip Select Number Active**

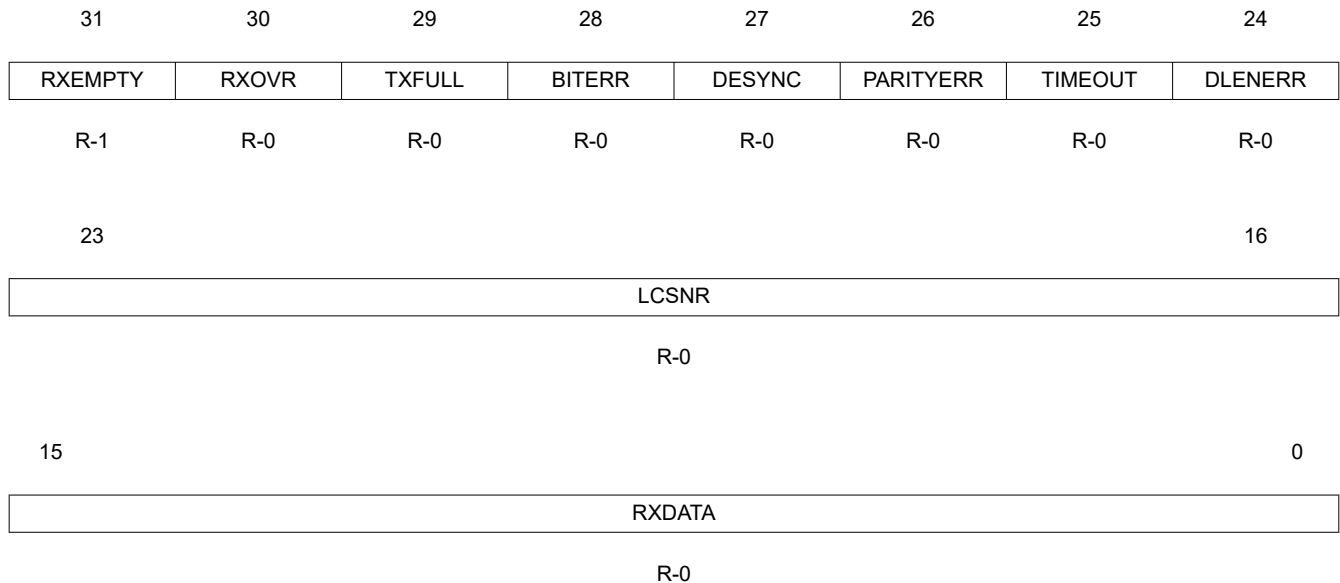
CSNR Value	Chip Select Active:						CSNR Value	Chip Select Active:					
	CS[5] <sup>(1)</sup>	CS[4] <sup>(1)</sup>	CS[3] <sup>(1)</sup>	CS[2] <sup>(1)</sup>	CS[1] <sup>(1)</sup>	CS[0]		CS[5] <sup>(1)</sup>	CS[4] <sup>(1)</sup>	CS[3] <sup>(1)</sup>	CS[2] <sup>(1)</sup>	CS[1] <sup>(1)</sup>	CS[0]
0h	No chip select pin is active.						20h	x					
1h						x	21h	x					x
2h					x		22h	x				x	
3h					x	x	23h	x				x	x
4h				x			24h	x			x		
5h				x		x	25h	x			x		x
6h				x	x		26h	x			x	x	
7h				x	x	x	27h	x			x	x	x
8h			x				28h	x		x			
9h			x			x	29h	x		x			x
Ah			x		x		2Ah	x		x		x	
Bh			x		x	x	2Bh	x		x		x	x
Ch			x	x			2Ch	x		x	x		
Dh			x	x		x	2Dh	x		x	x		x
Eh			x	x	x		2Eh	x		x	x	x	
Fh			x	x	x	x	2Fh	x		x	x	x	x
10h		x					30h	x	x				
11h		x				x	31h	x	x				x
12h		x			x		32h	x	x			x	
13h		x			x	x	33h	x	x			x	x
14h		x		x			34h	x	x		x		
15h		x		x		x	35h	x	x		x		x
16h		x		x	x		36h	x	x		x	x	
17h		x		x	x	x	37h	x	x		x	x	x
18h		x	x				38h	x	x	x			
19h		x	x			x	39h	x	x	x			x
1Ah		x	x		x		3Ah	x	x	x		x	
1Bh		x	x		x	x	3Bh	x	x	x		x	x
1Ch		x	x	x			3Ch	x	x	x	x		
1Dh		x	x	x		x	3Dh	x	x	x	x		x
1Eh		x	x	x	x		3Eh	x	x	x	x	x	
1Fh		x	x	x	x	x	3Fh	x	x	x	x	x	x

(1) If your MibSPI does not have this chip select pin, this bit is 0.



### 12.8.17 SPI Receive Buffer Register (SPIBUF)

**Figure 12-42. SPI Receive Buffer Register (SPIBUF) [offset = 40h]**



LEGEND: R = Read only; -n = value after reset

**Table 12-25. SPI Receive Buffer Register (SPIBUF) Field Descriptions**

Bit	Field	Value	Description
31	RXEMPTY	<div>0</div> <div>1</div>	<p>Receive data buffer empty. When the host reads the RXDATA field or the entire SPIBUF register, it automatically sets the RXEMPTY flag. When a data transfer is completed, the received data is copied into RXDATA and the RXEMPTY flag is cleared.</p> <p>New data has been received and copied into RXDATA.</p> <p>No data has been received since the last read of RXDATA.</p> <p>This flag gets set to 1 under the following conditions:</p> <ul style="list-style-type: none"> <li>Reading the RXDATA field of the SPIBUF register</li> <li>Writing a 1 to clear the RXINTFLG bit in the SPI Flag Register (SPIFLG)</li> </ul> <p>Write-clearing the RXINTFLG bit before reading the SPIBUF indicates the received data is being ignored. Conversely, RXINTFLG can be cleared by reading the RXDATA field of SPIBUF (or the entire register).</p>
30	RXOVR	<div>0</div> <div>1</div>	<p>Receive data buffer overrun. When a data transfer is completed and the received data is copied into RXBUF while it is already full, RXOVR is set. Overruns always occur to RXBUF, not to SPIBUF; the contents of SPIBUF are overwritten only after it is read by the Peripheral (VBUSP) master (CPU, DMA, or other host processor).</p> <p>If enabled, the RXOVRN interrupt is generated when RXBUF is overwritten, and reading either SPI Flag Register (SPIFLG) or SPIVEXTx shows the RXOVRN condition. Two read operations from the SPIBUF register are required to reach the overwritten buffer word (one to read SPIBUF, which then transfers RXDATA into SPIBUF for the second read).</p> <p><b>Note:</b> This flag is cleared to 0 when the RXDATA field of the SPIBUF register is read.</p> <p><b>Note:</b> A special condition under which RXOVR flag gets set. If both SPIBUF and RXBUF are already full and while another buffer receive is underway, if any errors such as TIMEOUT, BITERR, and DLEN_ERR occur, then RXOVR in RXBUF and SPI Flag Register (SPIFLG) registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal receive overrun.</p> <p>No receive data overrun condition occurred since last read of the data field.</p> <p>A receive data overrun condition occurred since last read of the data field.</p>

**Table 12-25. SPI Receive Buffer Register (SPIBUF) Field Descriptions (continued)**

Bit	Field	Value	Description
29	TXFULL	0 1	<p>Transmit data buffer full. This flag is a read-only flag. Writing into the SPIDAT0 or SPIDAT1 field while the TX shift register is full will automatically set the TXFULL flag. Once the word is copied to the shift register, the TXFULL flag will be cleared. Writing to SPIDAT0 or SPIDAT1 when both TXBUF and the TX shift register are empty does not set the TXFULL flag.</p> <p>The transmit buffer is empty; SPIDAT0/SPIDAT1 is ready to accept a new data.</p> <p>The transmit buffer is full; SPIDAT0/SPIDAT1 is not ready to accept new data.</p>
28	BITERR	0 1	<p>Bit error. There was a mismatch of internal transmit data and transmitted data.</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the SPIBUF register is read.</b></p> <p>No bit error occurred.</p> <p>A bit error occurred. The SPI samples the signal of the transmit pins (master: SIMOx, slave: SOMIx) at the receive point (one-half clock cycle after the transmit point). If the sampled value differs from the transmitted value, a bit error is detected and the BITERR flag is set. Possible reasons for a bit error include noise, an excessively high bit rate, capacitive load, or another master/slave trying to transmit at the same time.</p>
27	DESYNC	0 1	<p>Desynchronization of slave device. This bit is valid in master mode only.</p> <p>The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus <math>t_{T2DELAY}</math>. If DESYNCENA is set, an interrupt is asserted. Desynchronization can occur if a slave device misses a clock edge coming from the master.</p> <p><b>Note: In the Compatibility Mode MibSPI, under some circumstances it is possible for a desync error detected for the previous buffer to be visible in the current buffer. This is because the receive completion flag/interrupt is generated when the buffer transfer is completed. But desynchronization is detected after the buffer transfer is completed. So, if the VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct desync condition. In multi-buffer mode, the desync flag is always assured to be for the current buffer.</b></p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the SPIBUF register is read.</b></p> <p>No slave desynchronization is detected.</p> <p>A slave device is desynchronized.</p>
26	PARITYERR	0 1	<p>Parity error. The calculated parity differs from the received parity bit.</p> <p>If the parity generator is enabled (selected individually for each buffer) an even or odd parity bit is added at the end of a data word. During reception of the data word, the parity generator calculates the reference parity and compares it to the received parity bit. If a mismatch is detected, the PARITYERR flag is set.</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the SPIBUF register is read.</b></p> <p>No parity error is detected.</p> <p>A parity error occurred.</p>
25	TIMEOUT	0 1	<p>Time-out because of non-activation of <math>\overline{\text{SPIEN}}_A</math> pin.</p> <p>The SPI generates a time-out when the slave does not respond in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected, the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition, the TIMEOUT flag in the status field of the corresponding buffer and in the SPI Flag Register (SPIFLG) is set.</p> <p><b>Note: This bit is valid only in master mode.</b></p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the SPIBUF register is read.</b></p> <p>No <math>\overline{\text{SPIEN}}_A</math> pin time-out occurred.</p> <p>An <math>\overline{\text{SPIEN}}_A</math> signal time-out occurred.</p>
24	DLENERR	0 1	<p>Data length error flag.</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the SPIBUF register is read.</b></p> <p>No data-length error occurred.</p> <p>A data length error occurred.</p>
23-16	LCSNR	0-FFh	<p>Last chip select number. LCSNR in the status field is a copy of CSNR in the corresponding control field. It contains the chip select number that was activated during the last word transfer.</p>

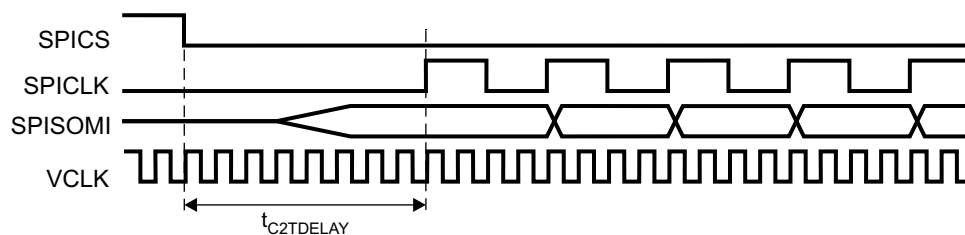
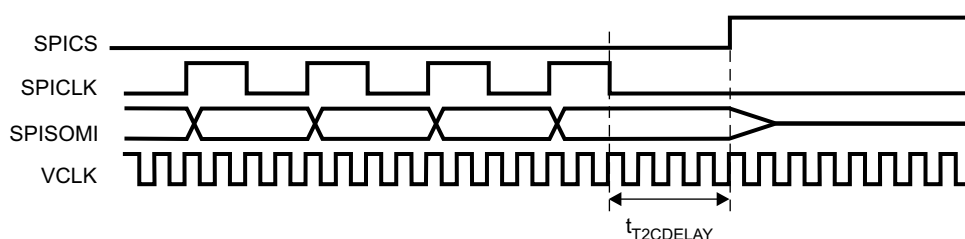
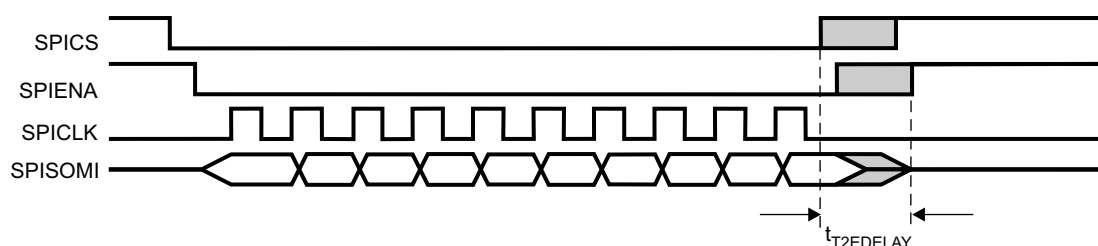
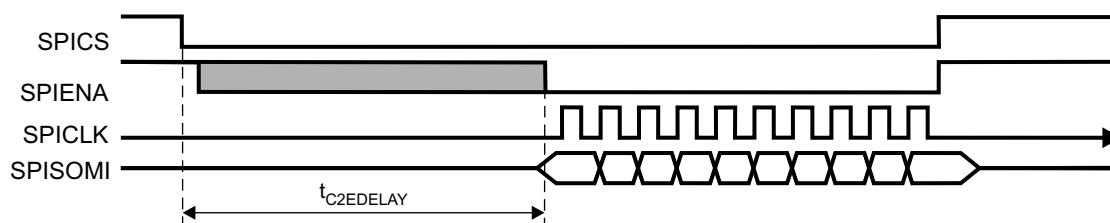
**Table 12-25. SPI Receive Buffer Register (SPIBUF) Field Descriptions (continued)**

Bit	Field	Value	Description
15-0	RXDATA	0-FFFFh	SPI receive data. This is the received word, transferred from the receive shift-register at the end of a transfer. Regardless of the programmed character length and the direction of shifting, the received data is stored right-justified in the register.



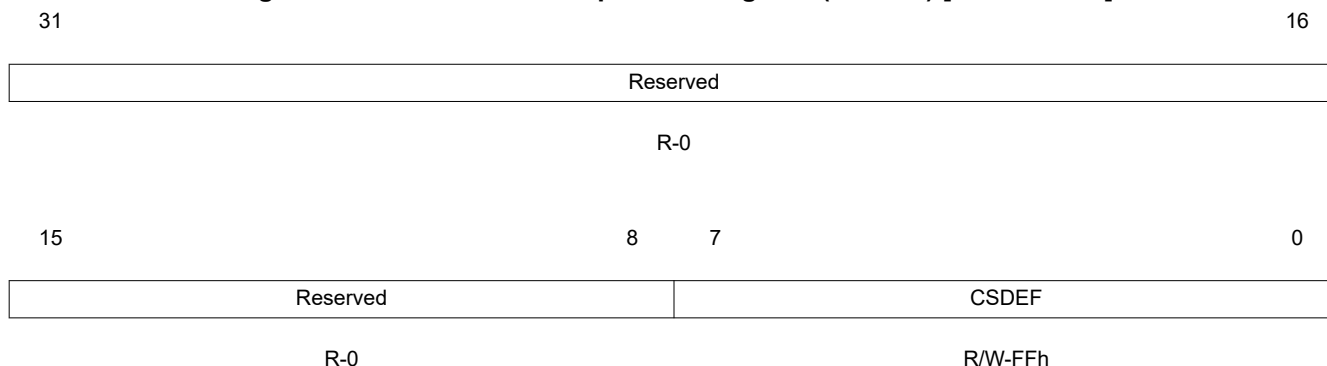
**Table 12-27. SPI Delay Register (SPIDELAY) Field Descriptions (continued)**

Bit	Field	Value	Description
23-16	T2CDELAY	0-FFh	<p>Transmit-end-to-chip-select-inactive-delay. See <a href="#">Figure 12-46</a> for an example. T2CDELAY is used only in master mode. It defines a hold time for the slave device that delays the chip select deactivation by a multiple of VCLK cycles after the last bit is transferred.</p> <p>The hold time value is calculated as follows:  <math>t_{T2CDELAY} = (T2CDELAY + 1) \times VCLK \text{ Period}</math></p> <p>Example: VCLK = 25 MHz -&gt; VCLK Period = 40ns; T2CDELAY = 03h;  <math>&gt; t_{T2CDELAY} = 160 \text{ ns}</math></p> <p>After the last data bit (or parity bit) is being transferred the chip select signal is held active for 160 ns.</p> <p><b>Note: If phase = 0, then between the last edge of SPICLK and rise-edge of SPICS there will be an additional delay of 0.5 SPICLK period. This is as per the SPI protocol.</b></p> <p>Both C2TDELAY and T2CDELAY counters do not have any dependency on the <math>\overline{SPIEN}</math> pin value. Even if the <math>\overline{SPIEN}</math> pin is asserted by the slave, the master will continue to delay the start of SPICLK until the C2TDELAY counter overflows.</p> <p>Similarly, even if the <math>\overline{SPIEN}</math> pin is deasserted by the slave, the master will continue to hold the SPICS pins active until the T2CDELAY counter overflows. In this way, it is assured that the setup and hold times of the <math>\overline{SPICS}</math> pins are determined by the delay timers alone. To achieve better throughput, it should be ensured that these two timers are kept at the minimum possible values.</p>
15-8	T2EDELAY	0-FFh	<p>Transmit-data-finished to ENA-pin-inactive time-out. T2EDELAY is used in master mode only. It defines a time-out value as a multiple of SPI clock before <math>\overline{SPIEN}</math> signal has to become inactive and after <math>\overline{SPICS}</math> becomes inactive. SPICLK depends on which data format is selected. If the slave device is missing one or more clock edges, it becomes de-synchronized. In this case, although the master has finished the data transfer, the slave is still waiting for the missed clock pulses and the ENA signal is not disabled.</p> <p>The T2EDELAY defines a time-out value that triggers the DESYNC flag, if the <math>\overline{SPIEN}</math> signal is not deactivated in time. The DESYNC flag is set to indicate that the slave device did not de-assert its <math>\overline{SPIEN}</math> pin in time to acknowledge that it received all bits of the sent word. See <a href="#">Figure 12-47</a> for an example of this condition.</p> <p><b>Note: DESYNC is also set if the SPI detects a de-assertion of <math>\overline{SPIEN}</math> before the end of the transmission.</b></p> <p>The time-out value is calculated as follows:  <math>t_{T2EDELAY} = T2EDELAY / SPIclock</math></p> <p>Example: SPIclock = 8 Mbit/s; T2EDELAY = 10h;  <math>&gt; t_{T2EDELAY} = 2 \mu s</math></p> <p>The slave device has to disable the ENA signal within 2 <math>\mu s</math>, otherwise DESYNC is set and an interrupt is asserted (if enabled).</p>
7-0	C2EDELAY	0-FFh	<p>Chip-select-active to ENA-signal-active time-out. C2EDELAY is used only in master mode and it applies only if the addressed slave generates an ENA signal as a hardware handshake response. C2EDELAY defines the maximum time between when the SPI activates the chip-select signal and the addressed slave has to respond by activating the ENA signal. C2EDELAY defines a time-out value as a multiple of SPI clocks. The SPI clock depends on whether data format 0 or data format 1 is selected. See <a href="#">Figure 12-48</a> for an example of this condition.</p> <p><b>Note: If the slave device does not respond with the ENA signal before the time-out value is reached, the TIMEOUT flag in the SPIFLG register is set and a interrupt is asserted (if enabled).</b></p> <p>If a time-out occurs, the SPI clears the transmit request of the timed-out buffer, sets the TIMEOUT flag for the current buffer, and continues with the transfer of the next buffer in the sequence that is enabled.</p> <p>The timeout value is calculated as follows:  <math>t_{C2EDELAY} = C2EDELAY / SPIclock</math></p> <p>Example: SPIclock = 8 Mbit/s; C2EDELAY = 30h;  <math>&gt; t_{C2EDELAY} = 6 \text{ ms}</math></p> <p>The slave device has to activate the ENA signal within 6 ms after the SPI has activated the chip select signal ( <math>\overline{SPICS}</math> ), otherwise the TIMEOUT flag is set and an interrupt is asserted (if enabled).</p>


**Figure 12-45. Example:  $t_{C2TDELAY} = 8$  VCLK Cycles**

**Figure 12-46. Example:  $t_{T2CDELAY} = 4$  VCLK Cycles**

**Figure 12-47. Transmit-Data-Finished-to-ENA-Inactive-Timeout**

**Figure 12-48. Chip-Select-Active-to-ENA-Signal-Active-Timeout**

## 12.8.20 SPI Default Chip Select Register (SPIDEF)

**Figure 12-49. SPI Default Chip Select Register (SPIDEF) [offset = 4Ch]**



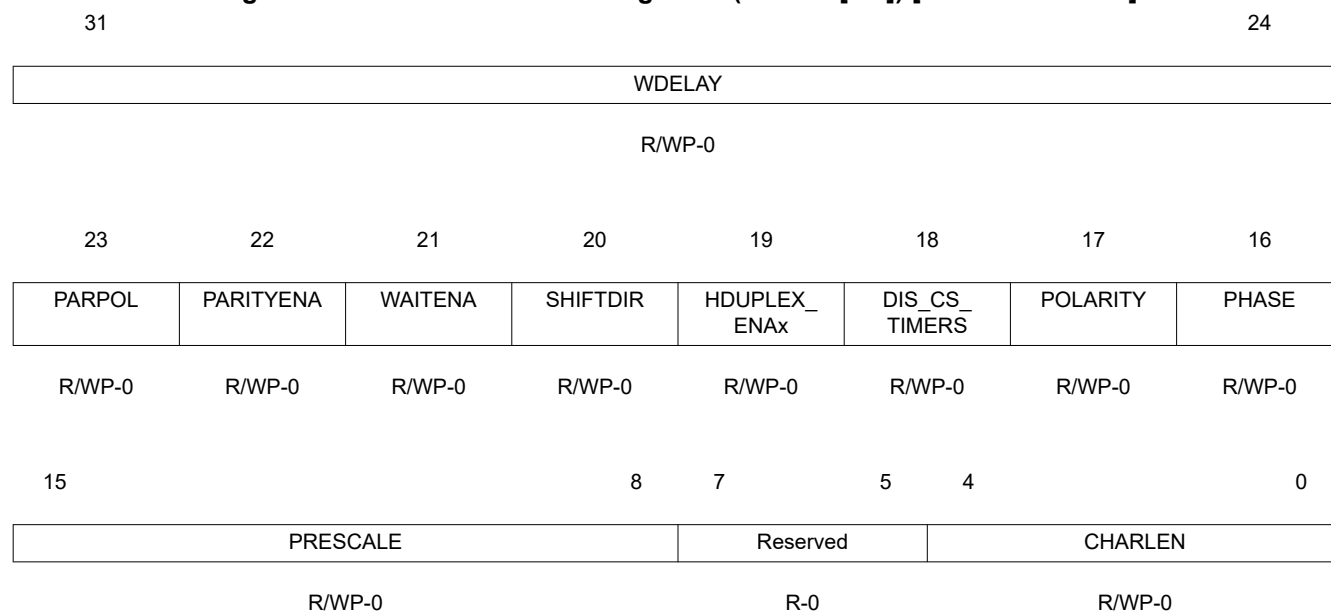
LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-28. SPI Default Chip Select Register (SPIDEF) Field Descriptions**

Bit	Field	Value	Description
31-8	Reserved	0	Reads return 0. Writes have no effect.
7-0	CDEF	0-FFh	Chip select default pattern. Master-mode only. The CSDEF bits are output to the $\overline{\text{SPICS}}$ pins when no transmission is being performed. It allows the user to set a programmable chip-select pattern that deselects all of the SPI slaves.
		0	$\overline{\text{SPICS}}$ is cleared to 0 when no transfer is active.
		1	$\overline{\text{SPICS}}$ is set to 1 when no transfer is active.

## 12.8.21 SPI Data Format Registers (SPIFMT)

**Figure 12-50. SPI Data Format Registers (SPIFMT[3:0]) [offset = 5Ch-50h]**



LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-29. SPI Data Format Registers (SPIFMT) Field Descriptions**

Bit	Field	Value	Description
31-24	WDELAY	0-FFh	Delay in between transmissions for data format x (x= 0,1,2,3).Idle time that will be applied at the end of the current transmission if the bit WDEL is set in the current buffer. The delay to be applied is equal to: $WDELAY \times P_{VCLK} + 2 \times P_{VCLK}$ $P_{VCLK} \rightarrow$ Period of VCLK.
23	PARPOL	0 1	Parity polarity: even or odd. PARPOLx can be modified in privilege mode only. It can be used for data format x (x= 0,1,2,3). 0 An even parity flag is added at the end of the transmit data stream. 1 An odd parity flag is added at the end of the transmit data stream.
22	PARITYENA	0 1	Parity enable for data format x. No parity generation/ verification is performed for this data format. 0 A parity bit is transmitted at the end of each transmitted word. At the end of a transfer the parity generator compares the received parity bit with the locally-calculated parity flag. If the parity bits do not match, the RXERR flag is set in the corresponding control field. The parity type (even or odd) can be selected via the PARPOL bit. 1 <b>Note: If an uncorrectable error flag is set in a slave-mode SPI, then the wrong parity bit will be transmitted to indicate to the master that there has been some issue with the data parity. The SPISOMI pins will be forced to transmit all 0s, and the parity bit will be transmitted as 1 if even parity is selected and as 0 if odd parity is selected (using the PARPOLx bit of this register). This behavior occurs regardless of an uncorrectable parity error on either TXRAM or RXRAM.</b>



**Table 12-29. SPI Data Format Registers (SPIFMT) Field Descriptions (continued)**

Bit	Field	Value	Description
21	WAITENA	0 1	The master waits for the ENA signal from slave for data format x. WAITENA is valid in master mode only. WAITENA enables a flexible SPI network where slaves with ENA signal and slaves without ENA signal can be mixed. WAITENA defines, for each transferred word, whether the addressed slave generates the ENA signal or not.  The SPI does not wait for the ENA signal from the slave and directly starts the transfer.  Before the SPI starts the data transfer it waits for the ENA signal to become low. If the ENA signal is not pulled down by the addressed slave before the internal time-out counter (C2EDELAY) overflows, then the master aborts the transfer and sets the TIMEOUT error flag.
20	SHIFTDIR	0 1	Shift direction for data format x. With bit SHIFTDIRx, the shift direction for data format x (x=0,1,2,3) can be selected.  0 MSB is shifted out first. 1 LSB is shifted out first.
19	HDUPLEX_ENAx	0 1	Half Duplex transfer mode enable for Data Format x. This bit controls the I/O function of SOMI/SIMO lines for a specific requirement where in the case of Master mode, TX pin - SIMO will act as an RX pin, and in the case of Slave mode, RX pin - SIMO will act as a TX pin.  0 Normal Full Duplex transfer. 1 If MASTER = 1, SPISIMO pin will act as an RX pin (No TX possible) If MASTER = 0, SPISIMO pin will act as a TX pin (No RX possible).  For all normal operations, HDUPLEX_ENAx bits should always remain 0. It is intended for the usage when the SPISIMO pin is used for both TX and RX operations at different times.
18	DIS CS TIMERS	0 1	Disable chip-select timers for this format. The C2TDELAY and T2CDELAY timers are by default enabled for all the data format registers. Using this bit, these timers can be disabled for a particular data format, if they are not required. When a master is handling multiple slaves, with varied set-up hold requirement, the application can selectively choose to include or not include the chip-select delay timers for any slaves.  0 Both C2TDELAY and T2CDELAY counts are inserted for the chip selects. 1 No C2TDELAY or T2CDELAY is inserted in the chip select timings.
17	POLARITY	0 1	SPI data format x clock polarity. POLARITYx defines the clock polarity of data format x. The following restrictions apply when switching clock phase and/or polarity: <ul style="list-style-type: none"> <li>In 3-pin/4-pin with <math>\overline{\text{SPIEN}}\text{A}</math> pin configuration of a slave SPI, the clock phase and polarity cannot be changed on-the-fly between two transfers. The slave should be reset and reconfigured if clock phase/polarity needs to be switched. In summary, SPI format switching is not fully supported in slave mode.</li> <li>Even while using chip select pins, the polarity of SPICLK can be switched only while the slave is not selected by a valid chip select. The master SPI should ensure that while switching SPICLK polarity, it has deselected all of its slaves. Otherwise, the switching of SPICLK polarity may be incorrectly treated as a clock edge by some slaves.</li> </ul> 0 If POLARITYx is cleared to 0, the SPI clock signal is low-inactive, that is, before and after data transfer the clock signal is low. 1 If POLARITYx is set to 1, the SPI clock signal is high-inactive, that is, before and after data transfer the clock signal is high.
16	PHASE	0 1	SPI data format x clock delay. PHASEx defines the clock delay of data format x.  0 If PHASEx is cleared to 0, the SPI clock signal is not delayed versus the transmit/receive data stream. The first data bit is transmitted with the first clock edge and the first bit is received with the second (inverse) clock edge.  1 If PHASEx is set to 1, the SPI clock signal is delayed by a half SPI clock cycle versus the transmit/receive data stream. The first transmit bit has to output prior to the first clock edge. The master and slave receive the first bit with the first edge.

**Table 12-29. SPI Data Format Registers (SPIFMT) Field Descriptions (continued)**

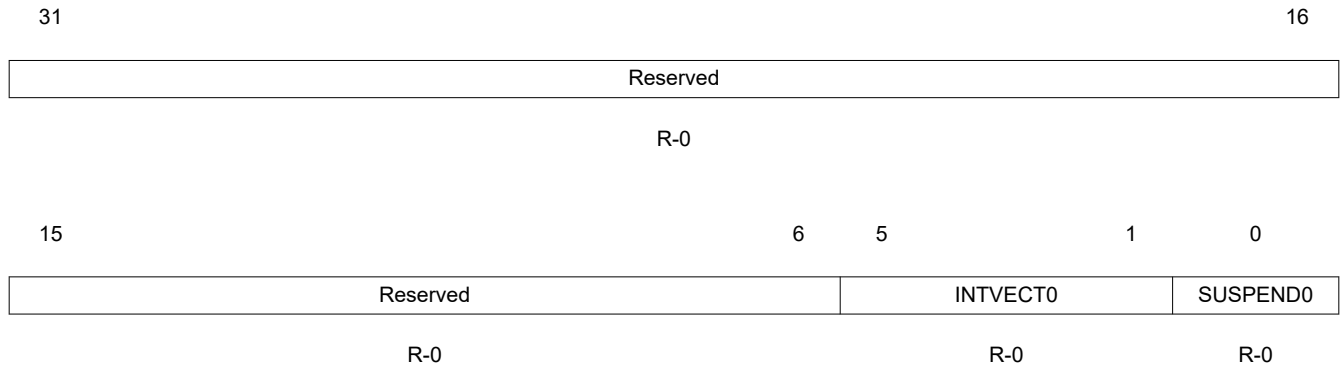
Bit	Field	Value	Description
15-8	PRESCALE		<p>SPI data format x prescaler. PRESCALE<sub>x</sub> determines the bit transfer rate of data format x if the SPI is the network master. PRESCALE<sub>x</sub> is use to derive SPICLK from VCLK. If the SPI is configured as slave, PRESCALE<sub>x</sub> <b>does not need</b> to be configured. The clock rate for data format x can be calculated as:</p> $BR_{Formatx} = VCLK / (PRESCALEx + 1)$ <p><b>Note: When PRESCALE<sub>x</sub> is cleared to 0, the SPI clock rate defaults to VCLK/2.</b></p>
7-5	Reserved	0	Reads return 0. Writes have no effect.
4-0	CHARLEN	0-1Fh	<p>SPI data format x data-word length. CHARLEN<sub>x</sub> defines the word length of data format x. Legal values are 0x02 (data word length = 2 bit) to 10h (data word length = 16). Illegal values, such as 00 or 1Fh are not allowed; their effect is indeterminate.</p>

## 12.8.22 Interrupt Vector 0 (INTVECT0)

### Note

The TG interrupt is not available in MibSPI in compatibility mode. Therefore, there is no possibility to access this register in compatibility mode.

**Figure 12-51. Interrupt Vector 0 (NTVECT0) [offset = 60h]**



LEGEND: R = Read only; -n = value after reset

**Table 12-30. Transfer Group Interrupt Vector 0 (INTVECT0)**

Bit	Field	Value	Description
31-6	Reserved	0	Reads return 0. Writes have no effect.
5-1	INTVECT0	<div>0</div> <div>1h + x</div> <div>11h</div> <div>13h</div> <div>12h</div> <div>14h</div> <div>All Other Combinations</div>	<p>INTVECT0. Interrupt vector for interrupt line INT0.</p> <p>Returns the vector of the pending interrupt at interrupt line INT0. If more than one interrupt is pending, INTVECT0 always references the highest prior interrupt source first.</p> <p><b>Note: This field reflects the status of the SPIFLG register in vector format. Any updates to the SPIFLG register will automatically cause updates to this field.</b></p> <p>There is no pending interrupt.</p> <p>Transfer group x (x = 0 to 15) has a pending interrupt. SUSPEND0 reflects the type of interrupt (<i>suspend</i> or <i>finished</i>).</p> <p>Error Interrupt pending. The lower half of SPIFLG contains more details about the type of error.</p> <p>The pending interrupt is a Receive Buffer Overrun interrupt.</p> <p><b>SPI mode:</b> The pending interrupt is a Receive Buffer Full interrupt.  <b>Mib mode:</b> Reserved. This bit combination should not occur.</p> <p><b>SPI mode:</b> The pending interrupt is a Transmit Buffer Empty interrupt.  <b>Mib mode:</b> Reserved. This bit combination should not occur.</p> <p><b>SPI mode:</b> Reserved. These bit combinations should not occur.</p>
0	SUSPEND0	<div>0</div> <div>1</div>	<p>Transfer suspended / Transfer finished interrupt flag.</p> <p>Every time INTVECT0 is read by the host, the corresponding interrupt flag of the referenced transfer group is cleared and INTVECT0 is updated with the vector coming next in the priority chain.</p> <p>The interrupt type is a transfer finished interrupt. In other words, the buffer array referenced by INTVECT0 has asserted an interrupt because all of data from the transfer group has been transferred.</p> <p>The interrupt type is a transfer suspended interrupt. In other words, the transfer group referenced by INTVECT0 has asserted an interrupt because the buffer to be transferred next is in suspend-to-wait mode.</p>

### Note

Reading from the INTVECT0 register when Transmit Empty is indicated does not clear the TXINTFLG flag in the SPI Flag Register (SPIFLG). Writing a new word to the SPIDATx register clears the Transmit Empty interrupt.

### Note

In multi-buffer mode, INTVECT0 contains the interrupt for the highest priority transfer group. A read from INTVECT0 automatically causes the next-highest priority transfer group's interrupt status to get loaded into INTVECT0 and its corresponding SUSPEND flag to get loaded into SUSPEND0. The transfer group with the lowest number has the highest priority, and the transfer group with the highest number has the lowest priority.

Reading the INTVECT0 register when the RXOVRN interrupt is indicated in multi-buffer mode does not clear the RXOVRN flag and hence does not clear the vector. The RXOVRN interrupt vector may be cleared in multi-buffer mode either by write-clearing the RXOVRN flag in the SPI Flag Register (SPIFLG) or by reading the RXRAM Overrun Buffer Address Register (RXOVRN\_BUF\_ADDR).

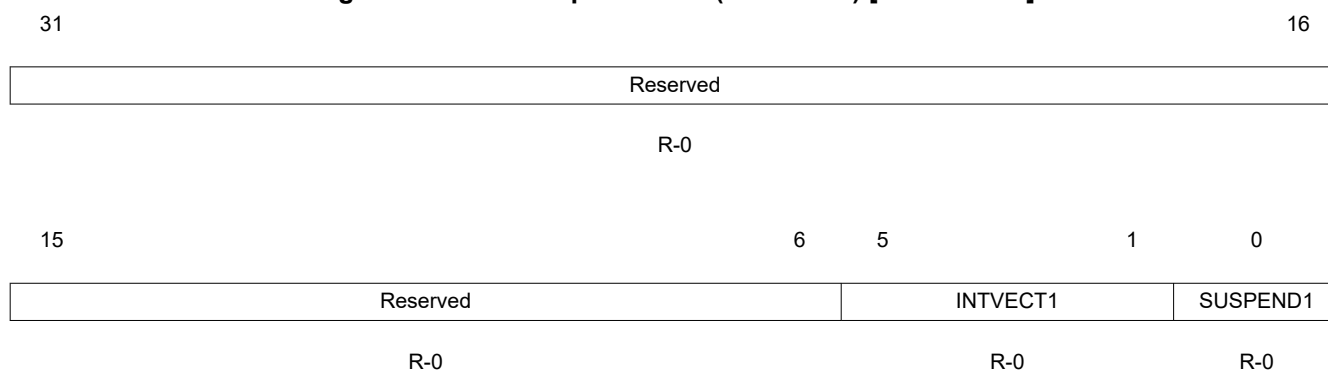
## 12.8.23

### 12.8.24 Interrupt Vector 1 (INTVECT1)

### Note

The TG interrupt is not available in SPI in compatibility mode compatibility mode. Therefore, there is no possibility to access this register in compatibility mode.

**Figure 12-52. Interrupt Vector 1 (INTVECT1) [offset = 64h]**



LEGEND: R = Read only; -n = value after reset

**Table 12-31. Transfer Group Interrupt Vector 1 (INTVECT1)**

Bit	Field	Value	Description
31-6	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-31. Transfer Group Interrupt Vector 1 (INTVECT1) (continued)**

Bit	Field	Value	Description
5-1	INTVECT1		INTVECT1. Interrupt vector for interrupt line INT1.  Returns the vector of the pending interrupt at interrupt line INT1. If more than one interrupt is pending, INTVECT1 always references the highest prior interrupt source first.  <b>Note: This field reflects the status of the SPIFLG register in vector format. Any updates to the SPIFLG register will automatically cause updates to this field.</b>
		0	There is no pending interrupt. <b>SPI mode only.</b>
		11h	Error Interrupt pending. The lower half of SPIINT1 contains more details about the type of error. <b>SPI mode only.</b>
		13h	The pending interrupt is a Receive Buffer Overrun interrupt. <b>SPI mode only.</b>
		12h	The pending interrupt is a Receive Buffer Full interrupt. <b>SPI mode only.</b>
		14h	The pending interrupt is a Transmit Buffer Empty interrupt. <b>SPI mode only.</b>
		All Other Combinations	Reserved. These bit combinations should not occur. <b>SPI mode only.</b>
0	SUSPEND1		Transfer suspended / Transfer finished interrupt flag.  Every time INTVECT1 is read by the host, the corresponding interrupt flag of the referenced transfer group is cleared and INTVECT1 is updated with the vector coming next in the priority chain.
		0	The interrupt type is a transfer finished interrupt. In other words, the buffer array referenced by INTVECT1 has asserted an interrupt because all of data from the transfer group has been transferred.
		1	The interrupt type is a transfer suspended interrupt. In other words, the transfer group referenced by INTVECT1 has asserted an interrupt because the buffer to be transferred next is in suspend-to-wait mode.

#### Note

Reading from the INTVECT1 register when Transmit Empty is indicated does not clear the TXINTFLG flag in the SPI Flag Register (SPIFLG). Writing a new word to the SPIDATx register clears the Transmit Empty interrupt.

#### Note

In multi-buffer mode, INTVECT1 contains the interrupt for the highest priority transfer group. A read from INTVECT1 automatically causes the next-highest priority transfer group's interrupt status to get loaded into INTVECT1 and its corresponding SUSPEND flag to get loaded into SUSPEND1. The transfer group with the lowest number has the highest priority, and the transfer group with the highest number has the lowest priority.

Reading the INTVECT1 register when the RXOVRN interrupt is indicated in multi-buffer mode does not clear the RXOVRN flag and hence does not clear the vector. The RXOVRN interrupt vector may be cleared in multi-buffer mode either by write-clearing the RXOVRN flag in the SPI Flag Register (SPIFLG) or by reading the RXOVRN\_BUF\_ADDR register.

### 12.8.25 SPI Pin Control Register 9 (SPIPC9)

SPIPC9 only applies to SPI2.

**Figure 12-53. SPI Pin Control Register 9 (SPIPC9) [offset = 68h]**

31	25	24	23	17	16
Reserved		SOMISRS0	Reserved		SIMOSRS0
R-0		R/W-0	R-0		R/W-0
15	12	11	10	9	8
15	12	11	10	9	8
Reserved		SOMISRS0	SIMOSRS0	CLKSRS	Reserved
R-0		R/W-0	R/W-0	R/W-0	R-0

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-32. SPI Pin Control Register 9 (SPIPC9) Field Descriptions**

Bit	Field	Value	Description
31-25	Reserved	0	Reads return the value that was last written. Writes have no effect.
24	SOMISRS0	0 1	SPI2 SOMI[0] slew control. This bit controls between the fast or slow slew mode. <b>Note: Duplicate Control Bits for SPI2 SOMI[0]. Bit 24 is not physically implemented. It is a mirror of bit 11. Any write to bit 24 will be reflected on bit 11. When bit 24 and bit 11 are simultaneously written, the value of bit 11 will control the SPI2 SOMI[0] pin. The read value of bit 24 always reflects the value of bit 11.</b> Fast mode is enabled; the normal output buffer is used for this pin. Slow mode is enabled; slew rate control is used for this pin.
23-17	Reserved	0	Reads return the value that was last written. Writes have no effect.
16	SIMOSRS0	0 1	SPI2 SIMO[0] slew control. This bit controls between the fast or slow slew mode. <b>Note: Duplicate Control Bits for SPI2 SIMO[0]. Bit 16 is not physically implemented. It is a mirror of bit 10. Any write to bit 16 will be reflected on bit 10. When bit 16 and bit 10 are simultaneously written, the value of bit 10 will control the SPI2 SIMO[0] pin. The read value of bit 16 always reflects the value of bit 10.</b> Fast mode is enabled; the normal output buffer is used for this pin. Slow mode is enabled; slew rate control is used for this pin.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11	SOMISRS0	0 1	SPI2 SOMI[0] slew control. This bit controls between the fast or slow slew mode. Fast mode is enabled; the normal output buffer is used for this pin. Slow mode is enabled; slew rate control is used for this pin.
10	SIMOSRS0	0 1	SPI2 SIMO[0] slew control. This bit controls between the fast or slow slew mode. Fast mode is enabled; the normal output buffer is used for this pin. Slow mode is enabled; slew rate control is used for this pin.
9	CLKSRS	0 1	SPI2 CLK slew control. This bit controls between the fast or slow slew mode. Fast mode is enabled; the normal output buffer is used for this pin. Slow mode is enabled; slew rate control is used for this pin.
8-0	Reserved	0	Reads return the value that was last written. Writes have no effect.

## 12.8.26 Parallel/Modulo Mode Control Register (SPIPMCTRL)

### Note

Do not configure MODCLKPOLx and MMODEx bits since this device does not support modulo mode.

### Note

The bits of this register are used in conjunction with the SPIFMTx registers. Each byte of this register corresponds to one of the SPIFMTx registers.

1. Byte0 (Bits 7:0) are used when SPIFMT0 register is selected by DFSEL[1:0] = 00 in the control field of a buffer.
2. Byte1 (Bits 15:8) are used when SPIFMT1 register is selected by DFSEL[1:0] = 01 in the control field of a buffer.
3. Byte2 (Bits 23:16) are used when SPIFMT2 register is selected by DFSEL[1:0] = 10 in the control field of a buffer.
4. Byte3 (Bits 31:24) are used when SPIFMT3 register is selected by DFSEL[1:0] = 11 in the control field of a buffer.

**Figure 12-54. Parallel/Modulo Mode Control Register (SPIPMCTRL) [offset = 6Ch]**

31	30	29	28	26	25	24
Reserved		MODCLKPOL3		MMODE3		PMODE3
R-0		R/WP-0		R/WP-0		R/WP-0
23	22	21	20	18	17	16
Reserved		MODCLKPOL2		MMODE2		PMODE2
R-0		R/WP-0		R/WP-0		R/WP-0
15	14	13	12	10	9	8
Reserved		MODCLKPOL1		MMODE1		PMODE1
R-0		R/WP-0		R/WP-0		R/WP-0
7	6	5	4	2	1	0
Reserved		MODCLKPOL0		MMODE0		PMODE0
R-0		R/WP-0		R/WP-0		R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-33. SPI Parallel/Modulo Mode Control Register (SPIPMCTRL) Field Descriptions**

Bit	Field	Value	Description
31-30	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-33. SPI Parallel/Modulo Mode Control Register (SPIPMCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
29	MODCLKPOL3	0 1	Modulo mode SPICLK polarity. This bit determines the polarity of the SPICLK in modulo mode only. If the MMODE3 bits are 000, this bit will be ignored. Normal SPICLK in all the modes. Polarity of the SPICLK will be inverted if Modulo mode is selected.
28-26	MMODE3	0 1h 2h 3h 4h 5h 6h-7h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if modulo option is supported by the module). Normal single data line mode - default (PMODE3 should be set to 00) 2-data line mode (PMODE3 should be set to 00) 3-data line mode (PMODE3 should be set to 00) 4-data line mode (PMODE3 should be set to 00) 5-data line mode (PMODE3 should be set to 00) 6-data line mode (PMODE3 should be set to 01) Reserved
25-24	PMODE3	0 1h 2h 3h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2, 4, or 8 data lines. Normal operation/1-data line (MMODE3 should be set to 000) 2-data line mode (MMODE3 should be set to 000) 4-data line mode (MMODE3 should be set to 000) 8-data line mode (MMODE3 should be set to 000)
23-22	Reserved	0	Reads return 0. Writes have no effect.
21	MODCLKPOL2	0 1	Modulo mode SPICLK polarity. This bit determines the polarity of the SPICLK in modulo mode only. If the MMODE2 bits are 000, this bit will be ignored. Normal SPICLK in all the modes. Polarity of the SPICLK will be inverted if Modulo mode is selected.
20-18	MMODE2	0 1h 2h 3h 4h 5h 6h-7h	These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if modulo option is supported by the module). 1-data line Mode - default (PMODE2 should be set to 00) 2-data line Mode (PMODE2 should be set to 00) 3-data line mode (PMODE2 should be set to 00) 4-data line mode (PMODE2 should be set to 00) 5-data line mode (PMODE2 should be set to 00) 6-data line mode (PMODE2 should be set to 01) Reserved
17-16	PMODE2	0 1h 2h 3h	Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2, 4, or 8 data lines. Normal operation/1-data line (MMODE2 should be set to 000) 2-data line mode (MMODE2 should be set to 000) 4-data line mode (MMODE2 should be set to 000) 8-data line mode (MMODE2 should be set to 000)
15-14	Reserved	0	Reads return 0. Writes have no effect.
13	MODCLKPOL1	0 1	Modulo mode SPICLK polarity. This bit determines the polarity of the SPICLK in modulo mode only. If the MMODE1 bits are 000, this bit will be ignored. Normal SPICLK in all the modes. Polarity of the SPICLK will be inverted if Modulo mode is selected.



**Table 12-33. SPI Parallel/Modulo Mode Control Register (SPIPMCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
12-10	MMODE1		These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if modulo option is supported by the module).
		0	1-data line mode - default (PMODE1 should be set to 00)
		1h	2-data line mode (PMODE1 should be set to 00)
		2h	3-data line mode (PMODE1 should be set to 00)
		3h	4-data line mode (PMODE1 should be set to 00)
		4h	5-data line mode (PMODE1 should be set to 00)
		5h	6-data line mode (PMODE1 should be set to 01)
		6h-7h	Reserved
9-8	PMODE1		Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2, 4, or 8 data lines.
		0	Normal operation/1-data line (MMODE1 should be set to 000)
		1h	2-data line mode (MMODE1 should be set to 000)
		2h	4-data line mode (MMODE1 should be set to 000)
		3h	8-data line mode (MMODE1 should be set to 000)
7-6	Reserved	0	Reads return 0. Writes have no effect.
5	MODCLKPOL0		Modulo mode SPICLK polarity. This bit determines the polarity of the SPICLK in modulo mode only. If the MMODE0 bits are 000, this bit will be ignored.
		0	Normal SPICLK in all the modes.
		1	Polarity of the SPICLK will be inverted if Modulo mode is selected.
4-2	MMODE0		These bits determine whether the SPI/MibSPI operates with 1, 2, 4, 5, or 6 data lines (if modulo option is supported by the module).
		0	1-data line mode - default (PMODE0 should be set to 00)
		1h	2-data line mode (PMODE0 should be set to 00)
		2h	3-data line mode (PMODE0 should be set to 00)
		3h	4-data line mode (PMODE0 should be set to 00)
		4h	5-data line mode (PMODE0 should be set to 00)
		5h	6-data line mode (PMODE0 should be set to 01)
		6h-7h	Reserved
1-0	PMODE0		Parallel mode bits determine whether the SPI/MibSPI operates with 1, 2, 4, or 8 data lines.
		0	Normal operation/1-data line (MMODE0 should be set to 000)
		1h	2-data line mode (MMODE0 should be set to 000)
		2h	4-data line mode (MMODE0 should be set to 000)
		3h	8-data line mode (MMODE0 should be set to 000)

## 12.8.27 Multi-buffer Mode Enable Register (MIBSPIE)

### Note

#### Accessibility of Multi-Buffer RAM

The multi-buffer RAM is not accessible unless the MSPIENA bit set to 1. The only exception to this is in test mode, where, by setting RXRAMACCESS to 1, the multi-buffer RAM can be fully accessed for both read and write.

**Figure 12-55. Multi-buffer Mode Enable Register (MIBSPIE) [offset = 70h]**

31		17	16
Reserved			RXRAM_ACCESS
	R-0		R/WP-0
15		1	0
Reserved			MSPIENA
	R-0		R/WP-0

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-34. Multi-buffer Mode Enable Register (MIBSPIE) Field Descriptions**

Bit	Field	Value	Description
31-17	Reserved	0	Reads return 0. Writes have no effect.
16	RXRAM ACCESS	0 1	Receive-RAM access control. During normal operating mode of SPI, the receive data/status portion of multi-buffer RAM is read-only. To enable testing of receive RAM, direct read/write access is enabled by setting this bit. The RX portion of multi-buffer RAM is not writable by the CPU. The whole of multi-buffer RAM is fully accessible for read/write by the CPU. <b>Note: The RX RAM ACCESS bit remains 0 after reset and it should remain cleared to 0 at all times, except when testing the RAM. SPI should be given a local reset by using the nRESET (SPIGCR0[0]) bit after RAM testing is performed so that the multi-buffer RAM gets re-initialized.</b>
15-1	Reserved	0	Reads return 0. Writes have no effect.
0	MSPIENA	0 1	Multi-buffer mode enable. After power-up or reset, MSPIENA remains cleared, which means that the SPI runs in compatibility mode by default. If multi-buffer mode is desired, this register should be configured first after configuring the SPIGCR0 register. If MSPIENA is not set to 1, the multi-buffer mode registers are not writable. The SPI runs in compatibility mode, that is, in this mode the MibSPI is fully code-compliant to the standard device SPI. No multi-buffered-mode features are supported. The SPI is configured to run in multi-buffer mode.

### Note

#### Accessibility of Registers

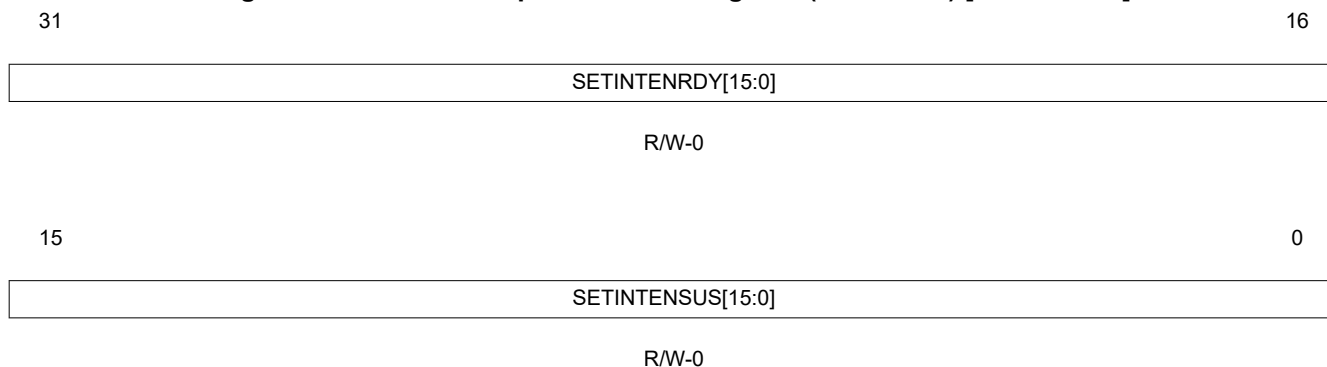
Registers from this offset address onwards are not accessible in SPI compatibility mode. They are accessible only in the multi-buffer mode.

### 12.8.28 TG Interrupt Enable Set Register (TGITENST)

The register TGITENST contains the TG interrupt enable flags for transfer-finished and for transfer-suspended events. Each of the enable bits in the higher half-word and the lower half-word of TGITENST belongs to one TG.

The register map shown in [Figure 12-56](#) and [Table 12-35](#) represents a super-set device with the maximum number of TGs (16) assumed. The actual number of bits available varies per device.

**Figure 12-56. TG Interrupt Enable Set Register (TGITENST) [offset = 74h]**



LEGEND: R/W = Read/Write; -n = value after reset

**Table 12-35. TG Interrupt Enable Set Register (TGITENST) Field Descriptions**

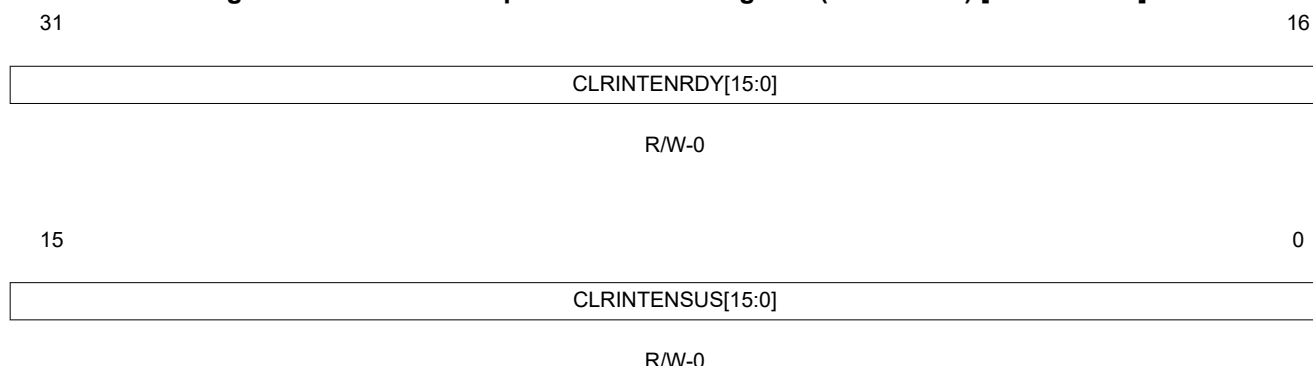
Bit	Field	Value	Description
31-16	SETINTENRDY[n]	0	TG interrupt set (enable) when transfer finished. Bit 16 corresponds to TG0, bit 17 corresponds to TG1, and so on. Read: The TGx-completed interrupt is disabled. This interrupt does not get generated when TGx completes. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-completed interrupt is enabled. The interrupt gets generated when TGx completes. Write: Enable the TGx-completed interrupt. The interrupt gets generated when TGx completes.
15-0	SETINTENSUS[n]	0	TG interrupt set (enabled) when transfer suspended. Bit 0 corresponds to TG0, bit 1 corresponds to TG1, and so on. Read: The TGx-completed interrupt is disabled. This interrupt does not get generated when TGx is suspended. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-completed interrupt is enabled. The interrupt gets generated when TGx is suspended. Write: Enable the TGx-completed interrupt. The interrupt gets generated when TGx is suspended.

## 12.8.29 TG Interrupt Enable Clear Register (TGITENCR)

The register TGITENCR is used to clear the interrupt enables for the TG-completed interrupt and the TG-suspended interrupts.

The register map shown in [Figure 12-57](#) and [Table 12-36](#) represents a super-set device with the maximum number of TGs (16) assumed. The actual number of bits available varies per device.

**Figure 12-57. TG Interrupt Enable Clear Register (TGITENCR) [offset = 78h]**



LEGEND: R/W = Read/Write; -n = value after reset

**Table 12-36. TG Interrupt Enable Clear Register (TGITENCR) Field Descriptions**

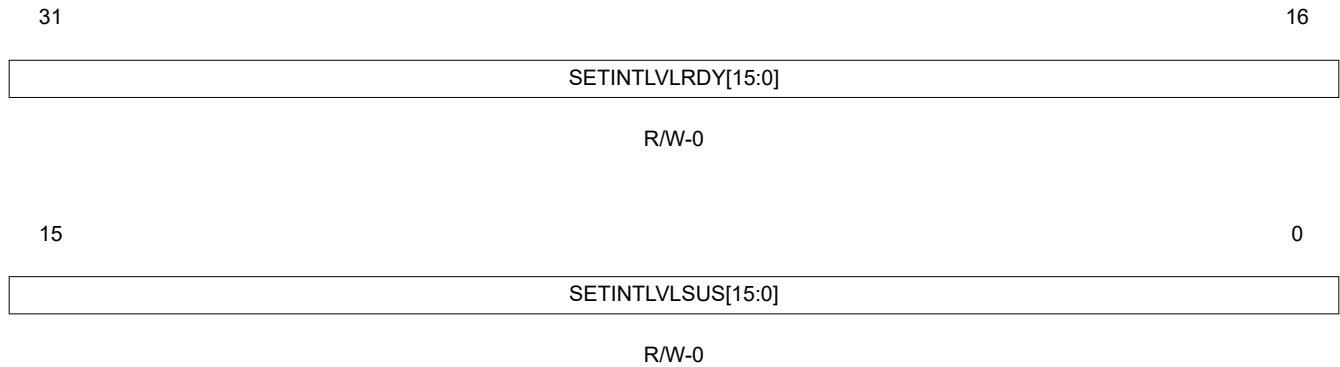
Bit	Field	Value	Description
31-16	CLRINTENRDY[n]	0	TG interrupt clear (disabled) when transfer finished. Bit 16 corresponds to TG0, bit 17 corresponds to TG1, and so on. Read: The TGx-completed interrupt is disabled. This interrupt does not get generated when TGx completes. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-completed interrupt is enabled. The interrupt gets generated when TGx completes. Write: Disable the TGx-completed interrupt. The interrupt does not get generated when TGx completes.
15-0	CLRINTENSUS[n]	0	TG interrupt clear (disabled) when transfer suspended. Bit 0 corresponds to TG0, bit 1 corresponds to TG1, and so on. Read: The TGx-completed interrupt is disabled. This interrupt does not get generated when TGx is suspended. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-completed interrupt is enabled. The interrupt gets generated when TGx is suspended. Write: Disable the TGx-completed interrupt. The interrupt does not get generated when TGx is suspended.

### 12.8.30 Transfer Group Interrupt Level Set Register (TGITLVST)

The register TGITLVST sets the level of interrupts for transfer completed interrupt and for transfer suspended interrupt to level 1.

The register map shown in [Figure 12-58](#) and [Table 12-37](#) represents a super-set device with the maximum number of TGs (16) assumed. The actual number of bits available varies per device.

**Figure 12-58. Transfer Group Interrupt Level Set Register (TGITLVST) [offset = 7Ch]**



LEGEND: R/W = Read/Write; -n = value after reset

**Table 12-37. Transfer Group Interrupt Level Set Register (TGITLVST) Field Descriptions**

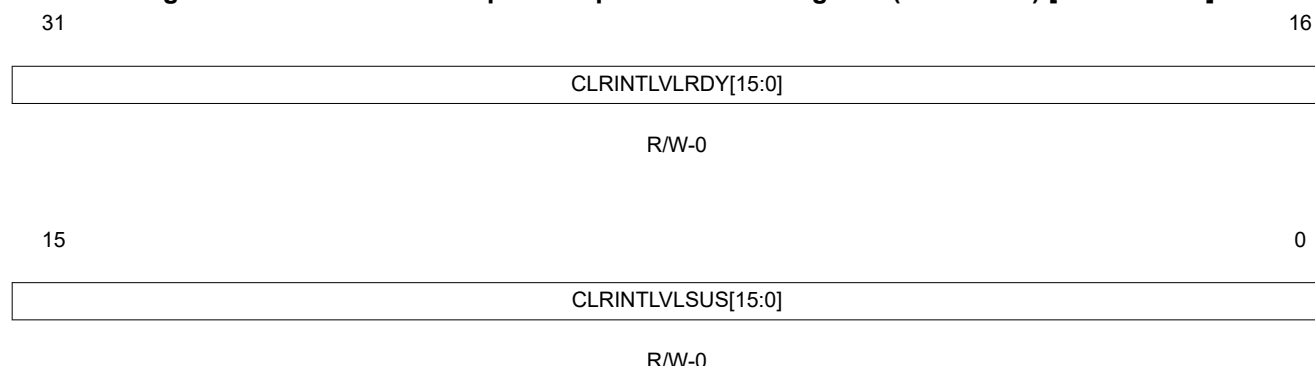
Bit	Field	Value	Description
31-16	SETINTLVLRDY[n]	0	Transfer-group completed interrupt level set. Bit 16 corresponds to TG0, bit 17 corresponds to TG1, and so on. Read: The TGx-completed interrupt is set to INT0. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-completed interrupt is set to INT1. Write: Set the TGx-completed interrupt to INT1.
15-0	SETINTLVLSUS[n]	0	Transfer-group suspended interrupt level set. Bit 0 corresponds to TG0, bit 1 corresponds to TG1, and so on. Read: The TGx-suspended interrupt is set to INT0. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-suspended interrupt is set to INT1. Write: Set the TGx-suspended interrupt to INT1.

### 12.8.31 Transfer Group Interrupt Level Clear Register (TGITLVCR)

The register TGITLVCR clears the level of interrupts for transfer completed interrupt and for transfer suspended interrupt to level 0.

The register map shown in [Figure 12-59](#) and [Table 12-38](#) represents a super-set device with the maximum number of TGs (16) assumed. The actual number of bits available varies per device.

**Figure 12-59. Transfer Group Interrupt Level Clear Register (TGITLVCR) [offset = 80h]**



LEGEND: R/W = Read/Write; -n = value after reset

**Table 12-38. Transfer Group Interrupt Level Clear Register (TGITLVCR) Field Descriptions**

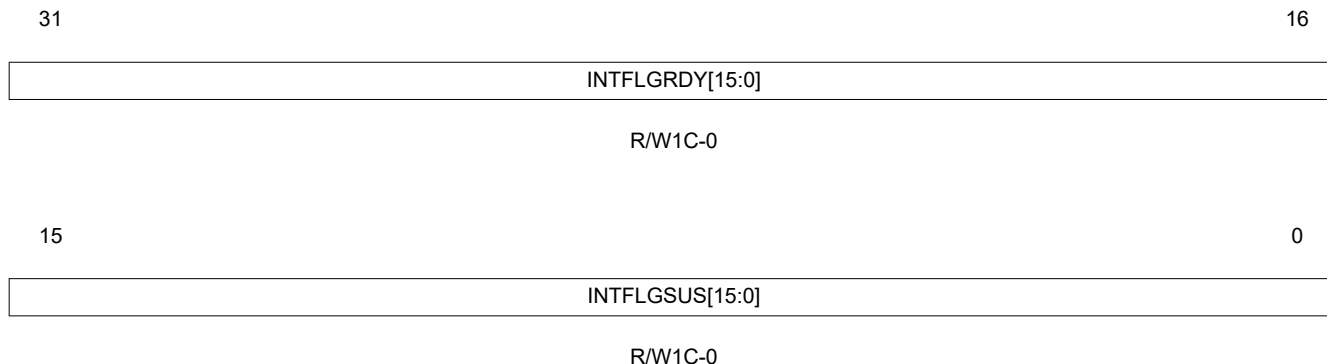
Bit	Field	Value	Description
31-16	CLRINTLVLRDY[n]	0	Transfer-group completed interrupt level clear. Bit 16 corresponds to TG0, bit 17 corresponds to TG1, and so on. Read: The TGx-completed interrupt is set to INT0. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-completed interrupt is set to INT1. Write: Clear the TGx-completed interrupt to INT0.
15-0	CLRINTLVLSUS[n]	0	Transfer group suspended interrupt level clear. Bit 0 corresponds to TG0, bit 1 corresponds to TG1, and so on. Read: The TGx-suspended interrupt is set to INT0. Write: A write of 0 to this bit has no effect.
		1	Read: The TGx-suspended interrupt is set to INT1. Write: Clear the TGx-suspended interrupt to INT0.

### 12.8.32 Transfer Group Interrupt Flag Register (TGINTFLG)

The TGINTFLG register comprises the transfer group interrupt flags for transfer-completed interrupts (INTFLGRDYx) and for transfer-suspended interrupts (INTFLGSUSx). Each of the interrupt flags in the higher half-word and the lower half-word of TGINTFLG belongs to one TG.

The register map shown in [Figure 12-60](#) and [Table 12-39](#) represents a super-set device with the maximum number of TGs (16) assumed. The actual number of bits available varies per device.

**Figure 12-60. Transfer Group Interrupt Flag Register (TGINTFLG) [offset = 84h]**



LEGEND: R/W = Read/Write; W1C = Write 1 to clear; -n = value after reset

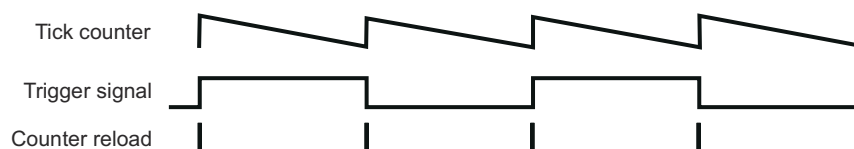
**Table 12-39. Transfer Group Interrupt Flag Register (TGINTFLG) Field Descriptions**

Bit	Field	Value	Description
31-16	INTFLGRDY[n]	0	Transfer-group interrupt flag for a transfer-completed interrupt. Bit 16 corresponds to TG0, bit 17 corresponds to TG1, and so on.  <b>Note: Read Clear Behavior.</b> Reading the interrupt vector registers TGINTVECT0 or TGINTVECT1 automatically clears the interrupt flag bit INTFLGRDYx referenced by the vector number given by INTVECT0/INTVECT1 bits, if the SUSPEND[0:1] bit in the vector registers is 0.  Read: No transfer-completed interrupt occurred since last clearing of the INTFLGRDYx flag. Write: A write of 0 to this bit has no effect.
		1	Read: A transfer finished interrupt from transfer group x occurred. No matter whether the interrupt is enabled or disabled (INTENRDYx = don't care) or whether the interrupt is mapped to INT0 or INT1, INTFLGRDYx is set right after the transfer from TGx is finished. Write: The corresponding bit flag is cleared.
15-0	INTFLGSUS[n]	0	Transfer-group interrupt flag for a transfer-suspend interrupt. Bit 0 corresponds to TG0, bit 1 corresponds to TG1, and so on.  <b>Note: Read Clear Behavior.</b> Reading the interrupt vector registers TGINTVECT0 or TGINTVECT1 automatically clears the interrupt flag bit INTFLGSUSx referenced by the vector number given by INTVECT0/INTVECT1 bits, if the SUSPEND[0:1] bit in the corresponding vector registers is 1.  Read: No transfer-suspended interrupt occurred since the last clearing of the INTFLGSUSx flag. Write: A write of 0 to this bit has no effect.
		1	Read: A transfer-suspended interrupt from TGx occurred. No matter whether the interrupt is enabled or disabled (INTENSUSx = don't care) or whether the interrupt is mapped to INT0 or INT1, INTFLGSUSx is set right after the transfer from transfer group x is suspended. Write: The corresponding bit flag is cleared.

### 12.8.33 Tick Count Register (TICKCNT)

One of the trigger sources for TGs is an internal periodic time trigger. This time trigger is called a tick counter and is basically a down-counter with a preload/reload value. Every time the tick counter detects an underflow it reloads the initial value and toggles the trigger signal provided to the TGs.

The trigger signal, shown in Figure 12-61 as a square wave, illustrates the different trigger event types for the TGs (for example, rising edge, falling edge, and both edges).



**Figure 12-61. Tick Counter Operation**

This register is shown in Figure 12-62 and described in Table 12-40.

**Figure 12-62. Tick Count Register (TICKCNT) [offset = 90h]**

31	30	29	28	27	16
TICKENA	RELOAD	CLKCTRL	Reserved		
R/W-0	R/S-0	R/W-0	R-0		
15					0
TICKVALUE					
R/W-0					

LEGEND: R/W = Read/Write; R = Read only; S = Set; -n = value after reset

**Table 12-40. Tick Count Register (TICKCNT) Field Descriptions**

Bit	Field	Value	Description
31	TICKENA	0	Tick counter enable. The internal tick counter is disabled. The counter value remains unchanged. <b>Note: When the tick counter is disabled, the trigger signal is forced low.</b>
		1	The internal tick counter is enabled and is clocked by the clock source selected by CLKCTRL. When TICKENA goes from 0 to 1, the tick counter is automatically loaded with the contents of TICKVALUE.
30	RELOAD		Pre-load the tick counter. RELOAD is a set-only bit; writing a 1 to it reloads the tick counter with the value stored in TICKVALUE. Reading RELOAD always returns a 0. <b>Note: When the tick counter is reloaded by the RELOAD bit, the trigger signal is not toggled.</b>
29-28	CLKCTRL	0	SPICLK of data word format 0 is selected as the clock source of the tick counter.
		1h	SPICLK of data word format 1 is selected as the clock source of the tick counter.
		2h	SPICLK of data word format 2 is selected as the clock source of the tick counter.
		3h	SPICLK of data word format 3 is selected as the clock source of the tick counter.
27-16	Reserved	0	Reads return 0. Writes have no effect.

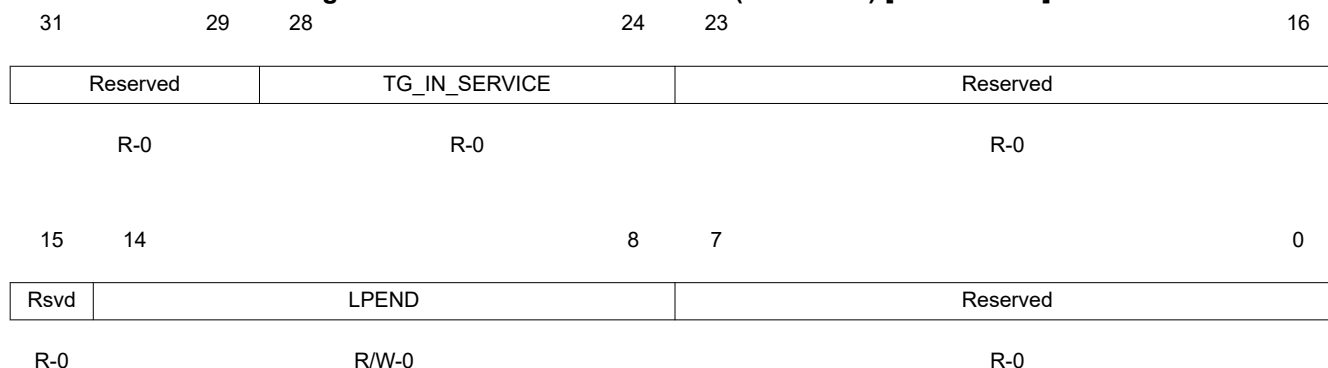


**Table 12-40. Tick Count Register (TICKCNT) Field Descriptions (continued)**

Bit	Field	Value	Description
15-0	TICKVALUE	0-FFFFh	Initial value for the tick counter. TICKVALUE stores the initial value for the tick counter. The tick counter is loaded with the contents of TICKVALUE every time an underflow condition occurs and every time the RELOAD flag is set by the host.

### 12.8.34 Last TG End Pointer (LTGPEND)

**Figure 12-63. Last TG End Pointer (LTGPEND) [offset = 94h]**



LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-41. Last TG End Pointer (LTGPEND) Field Descriptions**

Bit	Field	Value	Description
31-29	Reserved	0	Reads return 0. Writes have no effect.
28-24	TG_IN_SERVICE	0 1h : 10h 11h-1Fh	<p>The TG number currently being serviced by the sequencer. These bits indicate the current TG that is being serviced. This field can generally be used for code debugging.</p> <p>No TG is being serviced by the sequencer.</p> <p>TG0 is being serviced by the sequencer.</p> <p>:</p> <p>TG15 is being serviced by the sequencer.</p> <p><b>Note: The number of transfer groups varies by device.</b></p> <p>Invalid values.</p>
23-15	Reserved	0	Reads return 0. Writes have no effect.
14-8	LPEND	0-7Fh	<p>Last TG end pointer. Usually the TG end address (PEND) is inherently defined by the start value of the starting pointer of the subsequent TG (PSTART). The TG ends one word before the next TG starts (PEND[x] = PSTART[x+1] - 1). For a full configuration of MibSPI, the last TG has no subsequent TG, that is, no end address is defined. Therefore, LPEND has to be programmed to specify explicitly the end address of the last TG.</p> <p><b>Note: When using all 8 transfer groups, program the LPEND bits to define the end of the last transfer group. When using less than 8 transfer groups, leave the LPEND bits programmed to point to the end of the buffer and create a dummy transfer group that defines the end of your last intentional transfer group and occupies all the remaining buffer space.</b></p>
7-0	Reserved	0	Reads return 0. Writes have no effect.

### 12.8.35 TGx Control Registers (TGxCTRL)

Each TG can be configured via one dedicated control register. The register description shows one control register (x) that is identical for all TGs. For example, the control register for TG2 is named TG2CTRL and is located at *base address + 98h + 4 × 2*. The actual number of available control registers varies by device.

**Figure 12-64. MibSPI TG Control Registers (TGxCTRL) [offsets = 98h-D4h]**

31		30		29		28		27		24	
TGENA		ONESHOT		PRST		TGTD		Reserved			
R/W-0		R/W-0		R/W-0		R-0		R-0			
23						20		19		16	
TRIG EVT						TRIG SRC					
R/W-0						R/W-0					
15		14				8		7		6	
0											
Rsvd		PSTART						Rsvd		PCURRENT	
R-0				R/W-0				R-0		R-0	

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-42. TG Control Registers (TGxCTRL) Field Descriptions**

Bit	Field	Value	Description
31	TGENA	0 1	TGx enable. If the correct event (TRIGEV <sub>x</sub> ) occurs at the selected source (TRIGSRC <sub>x</sub> ), a group transfer is initiated if no higher-priority TG is in active-transfer mode or if one or more higher-priority TGs are in transfer-suspend mode. Disabling a TG while a transfer is ongoing will finish the ongoing word transfer but not the whole group transfer. TGx is disabled. TGx is enabled.
30	ONESHOT <sub>x</sub>	0 1	Single transfer for TGx. TGx initiates a transfer every time a trigger event occurs and TGENA is set. A transfer from TGx will be performed only once (one shot) after a valid trigger event at the selected trigger source. After the transfer is finished the TGENA <sub>x</sub> control bit will be cleared and therefore no additional transfer can be triggered before the host enables the TG again. This one shot mode ensures that after one group transfer the host has enough time to read the received data and to provide new transmit data.

**Table 12-42. TG Control Registers (TGxCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
29	PRSTx		<p>TGx pointer reset mode. Configures the way to resolve trigger events during an ongoing transfer. This bit is meaningful only for level-triggered TGs. Edge-triggered TGs cannot be restarted before their completion by another edge. The PRST bit will have no effect on this behavior.</p> <p><b>Note: When the PRST bit is set, if the buffer being transferred at the time of a new trigger event is a LOCK, CSHOLD or NOBRK buffer, then only after finishing those transfers, the TG will be restarted. This means that even if the TG is retriggered, the TG will only be restarted after finishing the transfer of the first non-LOCK or non-CSHOLD buffer. In the case of the NOBRK buffer, after completing the ICOUNT number of transfers, the TG will be restarted from its PSTART.</b></p> <p>This means that TX control fields such as LOCK and CSHOLD, and DMA control fields such as NOBRK have higher priority over anything else. They have the capability to delay the restart of the TG even if it is retriggered when PRST is 1.</p> <p>0 If a trigger event occurs during a transfer from TGx, the event is ignored and is not stored internally. The TGx transfer has priority over additional trigger events.</p> <p>1 The TGx pointer (PCURRENTx) will be reset to the start address (PSTARTx) when a valid trigger event occurs at the selected trigger source while a transfer from the same TG is ongoing. Every trigger event resets PCURRENTx no matter whether the concerned TG is in transfer mode or not. The trigger events have priority over the ongoing transfer.</p>
28	TGTDx	<p>0 TGx has not been triggered or is no longer waiting for service.</p> <p>1 TGx has been triggered and is either currently being serviced or waiting for servicing.</p>	
27-24	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-42. TG Control Registers (TGxCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
23-20	TRIGEVTx		Type of trigger event. A level-triggered TG can be stopped by de-activating the level trigger. However, the following restrictions apply. <ul style="list-style-type: none"> <li>Deactivating the level trigger for a TG during a NOBRK transfer does not stop the transfers until all of the ICOUNT number of buffers are transferred for the NOBRK buffer. Once a NOBRK buffer is prefetched, the trigger event loses control over the TG until the NOBRK buffer transfer is completed.</li> <li>Once the transfer of a buffer with CSHOLD or LOCK bit set starts, deactivating the trigger level does not stop the transfer until the sequencer completes the transfer of the next non-CSHOLD or non-LOCK buffer in the same TG.</li> <li>Once the last buffer in a TG is pre-fetched, de-activating the trigger level does not stop the transfer group until the last buffer transfer is completed. This means even if the trigger level is deactivated at the beginning of the penultimate (one-before-last) buffer transfer, the sequencer continues with the same TG until it is completed.</li> </ul>
		0	never Never trigger TGx. This is the default value after reset.
		1h	rising edge A rising edge (0 to 1) at the selected trigger source (TRIGSRCx) initiates a transfer for TGx
		2h	falling edge A falling edge (1 to 0) at the selected trigger source (TRIGSRCx) initiates a transfer for TGx
		3h	both edges Rising and falling edges at the selected trigger source (TRIGSRCx) initiates a transfer for TGx
		4h	Rsvd Reserved
		5h	high-active While the selected trigger source (TRIGSRCx) is at a logic-high level (1), the group transfer is continued and at the end of one group, transfer is restarted at the beginning. If the logic level changes to low (0) during an ongoing group transfer, the whole group transfer will be stopped. <b>Note: If ONESHOTx is set, the transfer is performed only once.</b>
		6h	low-active While the selected trigger source (TRIGSRCx) is at a logic-low level (0), the group transfer is continued and at the end of one group, transfer is restarted at the beginning. If the logic level changes to high (1) during an ongoing group transfer, the whole group transfer will be stopped. <b>Note: If ONESHOTx is set, the transfer is performed only once.</b>
		7h	always A repetitive group transfer will be performed. <b>Note: By setting the TRIGSRC to 0, the TRIGEVt to 7h (ALWAYS), and the ONESHOTx bit to 1, software can trigger this TG. Upon setting the TGENA bit, the TG is immediately triggered.</b> <b>Note: If ONESHOTx is set, the transfer is performed only once.</b>
		8h-Fh	Rsvd Reserved

**Table 12-42. TG Control Registers (TGxCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
19-16	TRIGSRCx	0	Disabled
		1h	EXT0 External trigger source 0. The actual source varies per device (for example, HET I/O channel, event pin).
		2h	EXT1 External trigger source 1. The actual source varies per device (for example, HET I/O channel, event pin).
		3h	EXT2 External trigger source 2. The actual source varies per device (for example, HET I/O channel, event pin).
		4h	EXT3 External trigger source 3. The actual source varies per device (for example, HET I/O channel, event pin).
		5h	EXT4 External trigger source 4. The actual source varies per device (for example, HET I/O channel, event pin).
		6h	EXT5 External trigger source 5. The actual source varies per device (for example, HET I/O channel, event pin).
		7h	EXT6 External trigger source 6. The actual source varies per device (for example, HET I/O channel, event pin).
		8h	EXT7 External trigger source 7. The actual source varies per device (for example, HET I/O channel, event pin).
		9h	EXT8 External trigger source 8. The actual source varies per device (for example, HET I/O channel, event pin).
		Ah	EXT9 External trigger source 9. The actual source varies per device (for example, HET I/O channel, event pin).
		Bh	EXT10 External trigger source 10. The actual source varies per device (for example, HET I/O channel, event pin).
		Ch	EXT11 External trigger source 11. The actual source varies per device (for example, HET I/O channel, event pin).
		Dh	EXT12 External trigger source 12. The actual source varies per device (for example, HET I/O channel, event pin).
		Eh	EXT13 External trigger source 13. The actual source varies per device (for example, HET I/O channel, event pin).
		Fh	TICK Internal periodic event trigger. The tick counter can initiate periodic group transfers.
15	Reserved	0	Reads return 0. Writes have no effect.
14-8	PSTARTx	0-7Fh	TG start address. PSTARTx stores the start address of the corresponding TG. The corresponding end address is inherently defined by the subsequent TG start address minus 1 ( $PENDx[TGx] = PSTARTx[TGx+1]-1$ ). PSTARTx is copied into PCURRENTx when: <ul style="list-style-type: none"> <li>The TG is enabled</li> <li>The end of the TG is reached during a transfer</li> <li>A trigger event occurs while PRST is set to 1</li> </ul>
7	Reserved	0	Reads return 0. Writes have no effect.
6-0	PCURRENTx	0-7Fh	Pointer to current buffer. PCURRENT is read-only. PCURRENTx stores the address (0...127) of the buffer that corresponds to this TG. If the TG switches from active-transfer mode to suspend-to-wait mode, PCURRENTx contains the address of the currently suspended word. After the TG resumes from suspend-to-wait mode, the next buffer will be transferred; that is, no buffer data is transferred because of suspend-to-wait mode.

---

**Note****Register bits vary by device**

TG0 has the highest priority and TG15 has the lowest priority. Under the following conditions, a lower-priority TG cannot be interrupted by a higher-priority TG:

1. When there is a CSHOLD or LOCK buffer, until the completion of the next buffer transfer that is a non-CSHOLD or non-LOCK buffer.
  2. An entire sequence of words transferred for a NOBRK DMA buffer.
  3. Once the last word in a TG is pre-fetched.
-

### 12.8.36 DMA Channel Control Register (DMAxCTRL)

Each DMA channel can be configured via one dedicated control register. The register description below shows one exemplary control register that is identical for all DMA channels; for example, the control register for DMA channel 0 is named DMA0CTRL. The MibSPI supports up to 8 bidirectional DMA channels.

The number of bidirectional DMA channels varies by device. The number of DMA channels and hence the number of DMA channel control registers may vary.

**Figure 12-65. DMA Channel Control Register (DMAxCTRL) [offset = D8h-F4h]**

31	30	24	23	20	19	16
ONESHOT	BUFID		RXDMA_MAP		TXDMA_MAP	
R/W-0		R/W-0		R/W-0		R/W-0
15	14	13	12	8		
RXDMAENA	TXDMAENA	NOBRK	ICOUNT			
R/W-0		R/W-0		R/W-0		
7	6	5	0			
Reserved	COUNT_BIT17	COUNT				
R-0	R-0	R-0				

LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-43. DMA Channel Control Register (DMAxCTRL) Field Descriptions**

Bit	Field	Value	Description
31	ONESHOT	<div>0</div> <div>1</div>	<p>Auto-disable of DMA channel after ICOUNT + 1 transfers.</p> <p><b>Note: This ONESHOT applies to the DMA channel identified by x and will autotisable based on ICOUNTx.</b></p> <p>The length of the block transfer is fully controlled by the DMA controller. The enable bits RXDMAENAx and TXDMAENAx are not modified by the MibSPI.</p> <p>ONESHOT allows a block transfer of defined length (ICOUNTx + 1), mainly controlled by the MibSPI and not by the DMA controller. After ICOUNTx + 1 transfers, the enable bits RXDMAENAx and TXDMAENAx are automatically cleared by the MibSPI, hence no more DMA requests are generated. In conjunction with NOBRKx, a burst transfer can be initiated without any other transfer through another buffer.</p>
30-24	BUFIDx	0-7Fh	Buffer utilized for DMA transfer. BUFIDx defines the buffer that is utilized for the DMA transfer. In order to synchronize the transfer with the DMA controller with the NOBRK condition the "suspend to wait until..." modes must be used.
23-20	RXDMA_MAPx	0-Fh	<p>Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data. RXDMA_MAPx defines the number of the physical DMA Request line that is connected to the receive path of the MibSPI DMA channel.</p> <p>If RXDMAENAx and TXDMAENAx are both set to 1, then RXDMA_MAPx shall differ from TXDMA_MAPx and shall differ from any other used physical DMA Request line. Otherwise, unexpected interference may occur.</p>



**Table 12-43. DMA Channel Control Register (DMAxCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
19-16	TXDMA_MAPx	0-Fh	Each MibSPI DMA channel can be linked to two physical DMA Request lines of the DMA controller. One request line for receive data and the other for request line for transmit data.  TXDMA_MAPx defines the number of the physical DMA Request line that is connected to the transmit path of the MibSPI DMA channel.  If RXDMAENAx and TXDMAENAx are both set to 1, then TXDMA_MAPx shall differ from RXDMA_MAPx and shall differ from any other used physical DMA Request line. Otherwise, unexpected interference may occur.
15	RXDMAENAx	0 1	Receive data DMA channel enable.  0 No DMA request upon new receive data.  1 The physical DMA channel for the receive path is enabled. The first DMA request pulse is generated after the first transfer from the referenced buffer (BUFIDx) is finished. The buffer should be configured in as "skip until RXEMPTY is set" or "suspend to wait until RXEMPTY is set" in order to ensure synchronization between the DMA controller and the MibSPI sequencer.
14	TXDMAENAx	0 1	Transmit data DMA channel enable.  0 No DMA request upon new transmit data.  1 The physical DMA channel for the transmit path is enabled. The first DMA request pulse is generated right after setting TXDMAENAx to load the first transmit data. The buffer should be configured in the as "skip until TXFULL is set" or "suspend to wait until TXFULL is set" in order to ensure synchronization between the DMA controller and the MibSPI sequencer.
13	NOBRKx	0 1	Non-interleaved DMA block transfer. This bit is available in master mode only.  <b>Note: Special Conditions during a NOBRK Buffer Transfer. If a NOBRK DMA buffer is currently being serviced by the sequencer, then it is not allowed to be disabled prematurely.</b>  During a NOBRK transfer, the following operations are not allowed: <ul style="list-style-type: none"> <li>• Clearing the NOBRKx bit to 0</li> <li>• Clearing the RXDMAENAx to 0 (if it is already 1)</li> <li>• Clearing the TXDMAENAx to 0 (if it is already 1)</li> <li>• Clearing the BUFMODE[2:0] bits in TXRAM to 000</li> </ul> <b>Note: Any attempts to perform these actions during a NOBRK transfer will produce unpredictable results.</b>  0 DMA transfers through the buffer referenced by BUFIDx are interleaved by data transfers from other active buffers or TGs. Every time the sequencer checks the DMA buffer, it performs one transfer and then steps to the next buffer.  1 NOBRKx ensures that ICOUNTx + 1 data transfers are performed from the buffer referenced by BUFIDx without a data transfer from any other buffer. The sequencer remains at the DMA buffer until ICOUNTx + 1 transfers have been processed. For example, this can be used to generate a burst transfer to one device without disabling the chip select signal in-between (the concerned buffer has to be configured with CSHOLD = 1). Another example would be to have a defined block data transfer in slave mode, synchronous to the master SPI.  <b>Note: Triggering of higher priority TGs or enabling of higher priority DMA channels will not interrupt a NOBRK block transfer.</b>
12-8	ICOUNTx	0-1Fh	Initial count of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits 0, it is reloaded with ICOUNTx. The real number of transfers equals ICOUNTx plus 1.  If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the DMA channels. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer. If ONESHOTx and NOBRKx are not set, ICOUNTx should be 0.  <b>Note: See Section 12.8.37 (ICOUNT) and Section 12.8.38 (DMACNTLEN) about how to increase the ICOUNT to a 16-bit value. With this extended capability, MibSPI can transfer a block of up to 65535 (65K) words without interleaving (if NOBRK is used) or without deasserting the chip select between the buffers (if CSHOLD is used).</b>
7	Reserved	0	Reads return 0. Writes have no effect.
6	COUNT_BIT17x		The 17th bit of the COUNT field of DMAxCOUNT register.

**Table 12-43. DMA Channel Control Register (DMAxCTRL) Field Descriptions (continued)**

Bit	Field	Value	Description
5-0	COUNTx	0-3Fh	<p>Actual number of remaining DMA transfers. This field contains the actual number of DMA transfers that remain, until the DMA channel is disabled, if ONESHOTx is set.</p> <p><b>Note: If the TX and RX DMA requests are enabled, the COUNT register will be decremented when the RX has been serviced.</b></p>

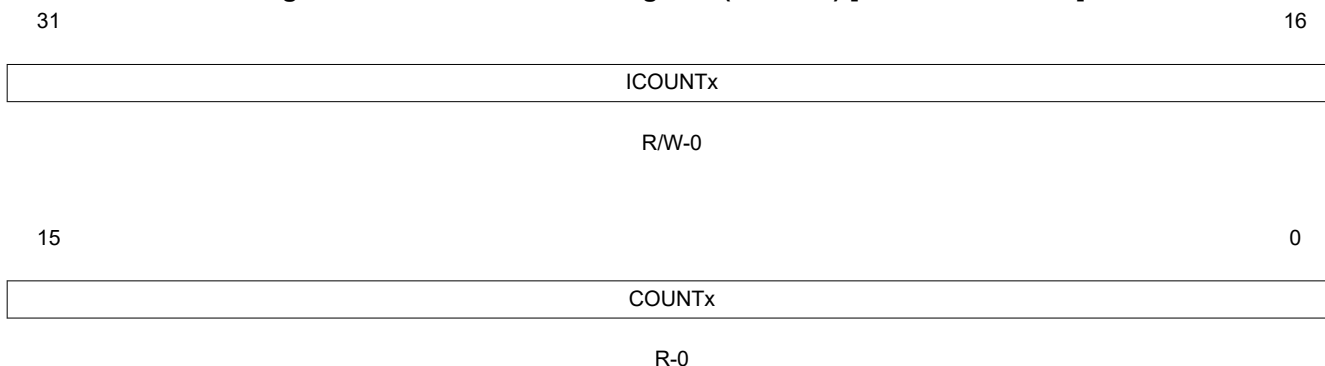
### 12.8.37 DMAxCOUNT Register (ICOUNT)

#### Note

These registers are used only if the LARGE COUNT bit in the DMACNTLEN register is set.

The number of bidirectional DMA channels varies by device. The number of DMA channels and hence the number of DMA registers varies by device.

**Figure 12-66. DMAxCOUNT Register (ICOUNT) [offset = F8h-114h]**



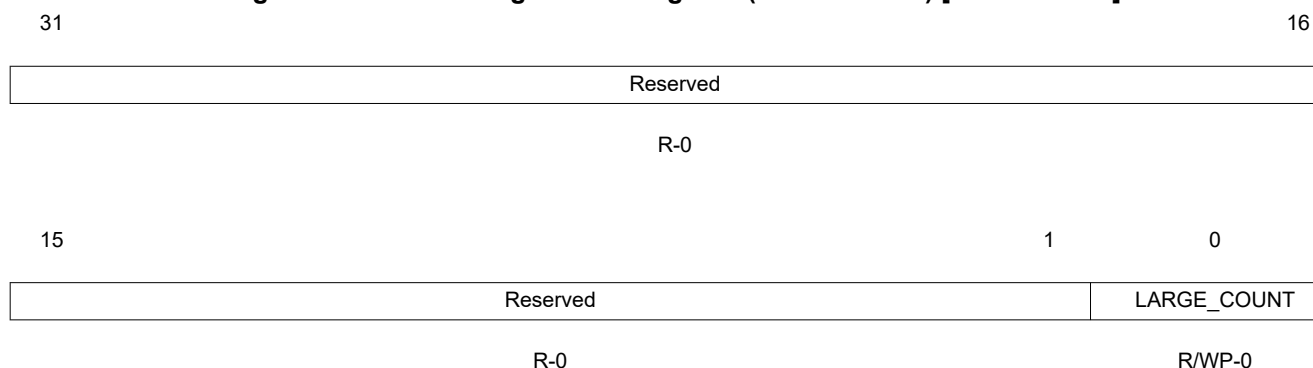
LEGEND: R/W = Read/Write; R = Read only; -n = value after reset

**Table 12-44. MibSPI DMAxCOUNT Register (ICOUNT) Field Descriptions**

Bit	Field	Value	Description
31-16	ICOUNTx	0-FFFFh	Initial number of DMA transfers. ICOUNTx is used to preset the transfer counter COUNTx. Every time COUNTx hits 0, it is reloaded with ICOUNTx. The real number of transfer equals ICOUNTx plus 1. If ONESHOTx is set, ICOUNTx defines the number of DMA transfers that are performed before the MibSPI automatically disables the corresponding DMA channel. If NOBRKx is set, ICOUNTx defines the number of DMA transfers that are performed in one sequence without a transfer from any other buffer.
15-0	COUNTx	0-FFFFh	Actual number of remaining DMA transfers. COUNTx Contains the actual number of DMA transfers that remain, until the DMA channel is disabled, if ONESHOTx is set. Since the real counter value is always ICOUNTx +1, the 17th bit of COUNTx is available on DMACTRLx[6] bit.  <b>Note: Usage Tip for Block Transfer Using a Single DMA Request.</b> It is possible to use the multi-buffer RAM to transfer chunks of data to/from an external SPI. A DMA Controller can be used to handle the data in bursts. Suppose a chunk of 64 bytes of data needs to be transferred and a single DMA request needs to be generated at the end of transferring the 64 bytes. This can be easily achieved by configuring a TG register for the 64 buffer locations and using the DMAxCTRL/DMAxCOUNT registers to configure the last buffer (64th) of the TG as the BUFID and enable RXDMA (NOBRK = 0). At the end of the transfer of the 64th buffer, a DMA request will be generated on the selected DMA request channel. The DMA controller can do a burst read of all 64 bytes from RXRAM and/or then do a burst write to all 64 bytes to the TXRAM for the next chunk.

### 12.8.38 DMA Large Count (DMACNTLEN)

**Figure 12-67. DMA Large Count Register (DMACNTLEN) [offset = 118h]**



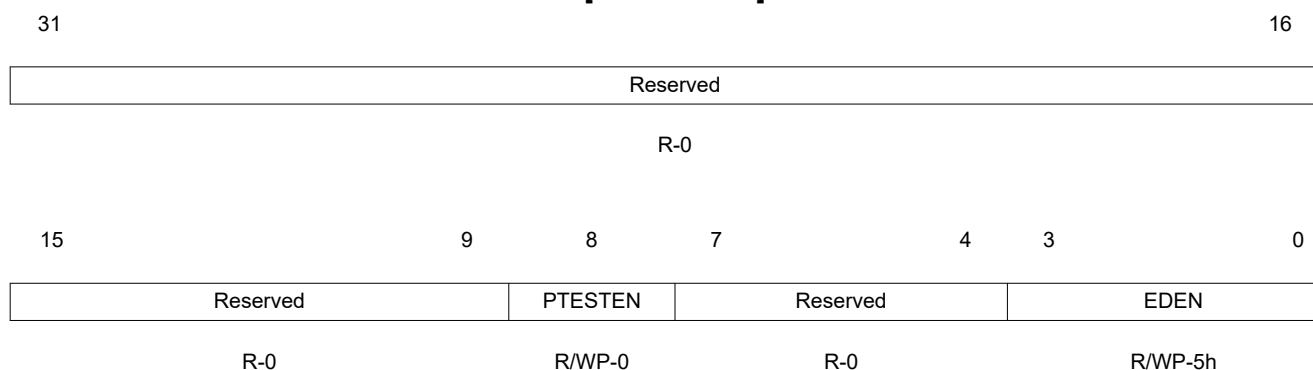
LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-45. MibSPI DMA Large Count Register (DMACNTLEN) Field Descriptions**

Bit	Field	Value	Description
31-1	Reserved	0	Reads return 0. Writes have no effect.
0	LARGE_COUNT	0	Select either the 16-bit DMAxCOUNT counters or the smaller counters in DMAxCTRL. Select the DMAxCTRL counters. Writes to the DMAxCTRL register will modify the ICOUNT value. Reading ICOUNT and COUNT can be done from the DMAxCTRL register. The DMAxCOUNT register should not be used since any write to this register will be overwritten by a subsequent write to the DMAxCTRL register to set the TXDMAENA or RXDMAENA bits.
		1	Select the DMAxCOUNT counters. Writes to the DMAxCTRL register will not modify the ICOUNT value. The ICOUNT value must be written to in the DMAxCOUNT register before the RXDMAENA or TXDMAENA bits are set in the DMAxCTRL register. The DMAxCOUNT register should be used for reading COUNT or ICOUNT.

### 12.8.39 Multi-buffer RAM Uncorrectable Parity Error Control Register (UERRCTRL)

**Figure 12-68. Multi-buffer RAM Uncorrectable Parity Error Control Register (UERRCTRL) [offset = 120h]**



LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-46. Multi-buffer RAM Uncorrectable Parity Error Control Register (UERRCTRL) Field Descriptions**

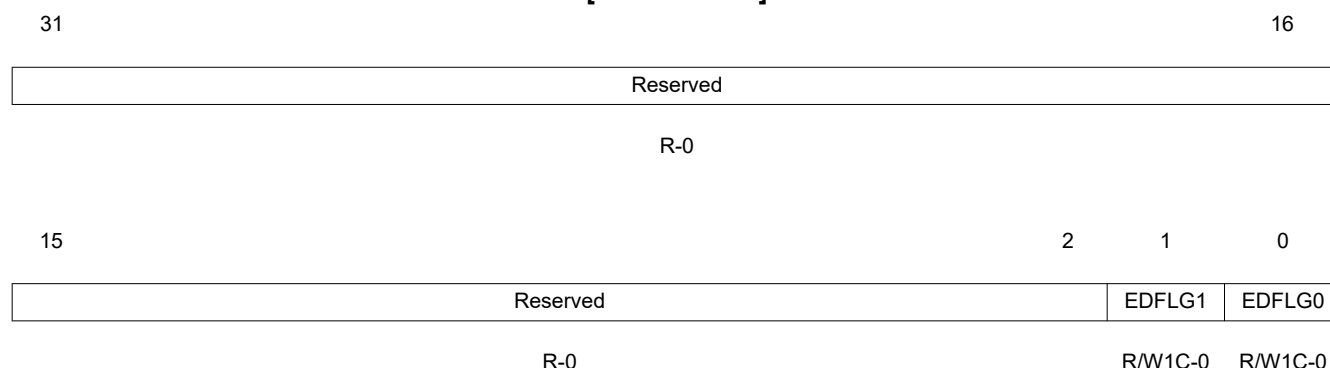
Bit	Field	Value	Description
31-9	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-46. Multi-buffer RAM Uncorrectable Parity Error Control Register (UERRCTRL)**  
**Field Descriptions (continued)**

Bit	Field	Value	Description
8	PTESTEN	0 1	Parity memory test enable. This bit maps the parity bits corresponding to multi-buffer RAM locations into the peripheral RAM frame to make them accessible by the CPU. See <a href="#">Section 12.10</a> for further details about parity memory testing. Parity bits are not memory-mapped. Parity bits are memory-mapped.
7-4	Reserved	0	Reads return 0. Writes have no effect.
3-0	EDEN	5h All Other Values	Error detection enable. These bits enable parity error detection. Parity error detection logic (default) is disabled. Parity error detection logic is enabled. <b>Note: It is recommended to write a 1010 to enable error detection, to guard against a soft error from disabling parity error detect</b>

### 12.8.40 Multi-buffer RAM Uncorrectable Parity Error Status Register (UERRSTAT)

**Figure 12-69. Multi-buffer RAM Uncorrectable Parity Error Status Register (UERRSTAT)**  
[offset = 124h]



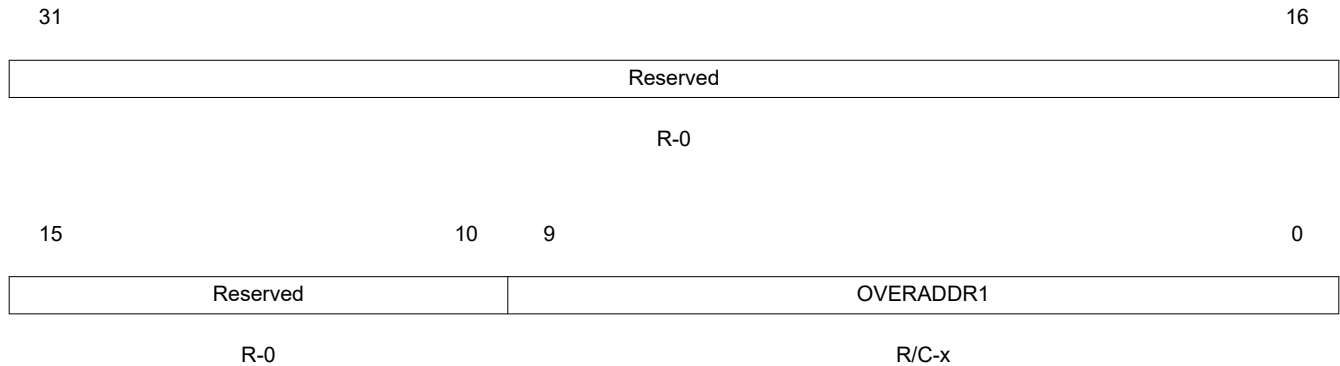
LEGEND: R/W = Read/Write; R = Read only; W1C = Write 1 to clear; -n = value after reset

**Table 12-47. Multi-buffer RAM Uncorrectable Parity Error Status Register (UERRSTAT)**  
**Field Descriptions**

Bit	Field	Value	Description
31-2	Reserved	0	Reads return 0. Writes have no effect.
1	EDFLG1	0	Uncorrectable parity error detection flag. This flag indicates if a parity error occurred in the RXRAM. <b>Note: Reading the UERRADDR1 register clears the EDFLG1 bit.</b> Read: No error has occurred. Write: Writing a 0 to this bit has no effect.
		1	Read: An error was detected and the address is captured in the UERRADDR1 register. Write: The bit is cleared to 0.
0	EDFLG0	0	Uncorrectable parity error detection flag. This flag indicates if a parity error occurred in the TXRAM. <b>Note: Reading the UERRADDR0 register clears the EDFLG0 bit.</b> Read: No error has occurred. Write: Writing a 0 to this bit has no effect.
		1	Read: An error was detected and the address is captured in the UERRADDR0 register. Write: The bit is cleared to 0.

### 12.8.41 RXRAM Uncorrectable Parity Error Address Register (UERRADDR1)

**Figure 12-70. RXRAM Uncorrectable Parity Error Address Register (UERRADDR1) [offset = 128h]**



LEGEND: R = Read only; C = Clear; -n = value after reset

**Table 12-48. RXRAM Uncorrectable Parity Error Address Register (UERRADDR1) Field Descriptions**

Bit	Field	Value	Description
31-10	Reserved	0	Reads return 0. Writes have no effect.
9-0	OVERADDR1	200h-3FFh	<p>Uncorrectable parity error address for RXRAM. This register holds the address where a parity error is generated while reading RXRAM. Only the CPU or DMA can read from RXRAM locations. The address captured is byte-aligned. This error address is frozen from being updated until it is read by the CPU. The offset address of RXRAM varies from 200h-3FFh.</p> <p>The register does not clear its contents during or after module-level reset, system-level reset or even power-on reset.</p> <p>A read operation to this register clears its contents to the default value 200h. After a power-on reset the contents will be unpredictable. A read operation can be performed after power-up to keep the register at its default value, if required. However, the contents of this register are meaningful only when EDFLG1 is set to 1.</p> <p><b>Note: A read of the UERRADDR1 register will clear EDFLG1 in the UERRSTAT register. However, in emulation mode when the SUSPEND signal is high, a read from the UERRADDR1 register does not clear EDFLG1.</b></p>

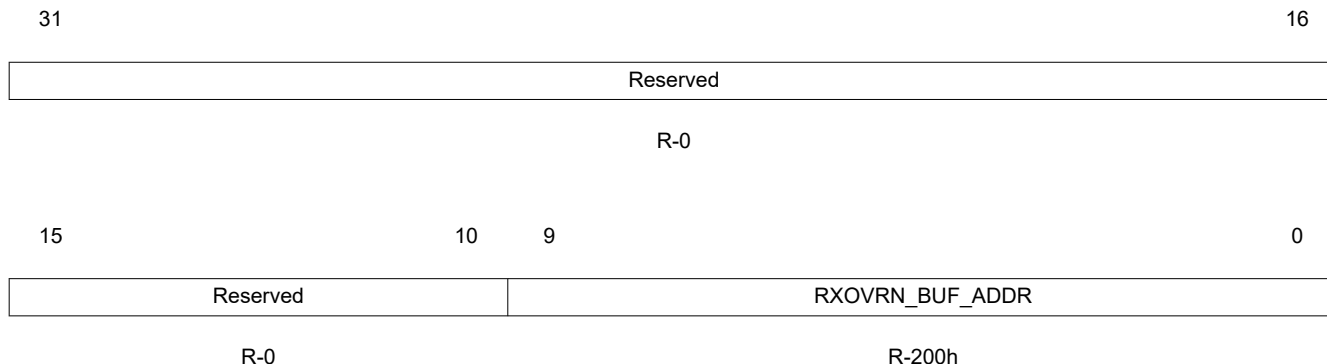




### 12.8.43 RXRAM Overrun Buffer Address Register (RXOVRN\_BUF\_ADDR)

In multi-buffer mode, if a particular RXRAM location is written by the MibSPI sequencer logic after the completion of a new transfer when that location already contains valid data, the RX\_OVR bit will be set to 1 while the data is being written. The RXOVRN\_BUF\_ADDR register captures the address of the RXRAM location for which a receiver overrun condition occurred.

**Figure 12-72. RXRAM Overrun Buffer Address Register (RXOVRN\_BUF\_ADDR) [offset = 130h]**



LEGEND: R = Read only; -n = value after reset

**Table 12-50. RXRAM Overrun Buffer Address Register (RXOVRN\_BUF\_ADDR) Field Descriptions**

Bit	Field	Value	Description
31-10	Reserved	0	Reads return 0. Writes have no effect.
9-0	RXOVRN_BUF_ADDR	200h-3FCh	<p>Address in RXRAM at which an overwrite occurred. This address value will show only the offset address of the RAM location in the multi-buffer RAM address space. Refer to the device-specific data sheet for the actual absolute address of RXRAM.</p> <p>This word-aligned address can vary from 200h-3FCh. Contents of this register are valid only when any of the INTVECT0 or INTVECT1 and SPIFLG registers show an RXOVRN error vector while in multi-buffer mode. If there are multiple overrun errors, then this register holds the address of first overrun address until it is read.</p> <p><b>Note:</b> Reading this register clears the RXOVRN interrupt flag in the SPIFLG register and the TGINTVECTx.</p> <p><b>Note:</b> Receiver overrun errors in multi-buffer mode can be completely avoided by using the SUSPEND until RXEMPTY feature, which can be programmed into each buffer of any TG. However, using the SUSPEND until RXEMPTY feature will make the sequencer wait until the current RXRAM location is read by the VBUS master before it can start the transfer for the same buffer location again. This may affect the overall throughput of the SPI transfer. By enabling the interrupt on RXOVRN in multi-buffer mode, the user can rely on interrupts to know if a receiver overrun has occurred. The address of the overrun in RXRAM is indicated in this RXOVRN_BUF_ADDR register.</p>



**Table 12-51. I/O-Loopback Test Control Register (IOLPBKTSTCR) Field Descriptions (continued)**

Bit	Field	Value	Description
19	CTRL DESYNC	0 1	Controls inducing of the desync error during I/O loopback test mode. Do not cause a desync error. Induce a desync error by forcing the incoming $\overline{\text{SPIEN}}\overline{\text{A}}$ pin (if functional) to remain 0 even after the transfer is complete. This forcing will be retained until the kernel reaches the idle state.
18	CTRL PARERR	0 1	Controls inducing of the parity errors during I/O loopback test mode. Do not cause a parity error. Induce a parity error by inverting the polarity of the parity bit.
17	CTRL TIMEOUT	0 1	Controls inducing of the timeout error during I/O loopback test mode. Do not cause a timeout error. Induce a timeout error by forcing the incoming $\overline{\text{SPIEN}}\overline{\text{A}}$ pin (if functional) to remain 1 when transmission is initiated. The forcing will be retained until the kernel reaches the idle state.
16	CTRL DLENERR	0 1	Controls inducing of the data length error during I/O loopback test mode. Do not cause a data-length error. Induce a data-length error. <i>Master mode:</i> The $\overline{\text{SPIEN}}\overline{\text{A}}$ pin (if functional) is forced to 1 when the module starts shifting data. <i>Slave mode:</i> The incoming $\overline{\text{SPICS}}$ pin (if functional) is forced to 1 when the module starts shifting data.
15-12	Reserved	0	Reads return 0. Writes have no effect.
11-8	IOLPBKSTENA	Ah All Other Values	Module I/O loopback test enable key. Enable I/O loopback test mode. Disable I/O loopback test mode.
7-6	Reserved	0	Reads return 0. Writes have no effect.
5-3	ERR SCS PIN	0 1h : 7h	Inject error on chip-select pin number x. The value in this field is decoded as the number of the chip select pin on which to inject an error. During analog loopback, if CTRL SCS PIN ERR bit is set to 1, then the chip select pin selected by this field is forced to the opposite of its value in the CSNR. Select $\overline{\text{SPICS}}[0]$ for injecting error. Select $\overline{\text{SPICS}}[1]$ for injecting error. : Select $\overline{\text{SPICS}}[7]$ for injecting error.
2	CTRL SCS PIN ERR	0 1	Enable/disable the injection of an error on the $\overline{\text{SPICS}}$ pins. The individual $\overline{\text{SPICS}}$ pins can be chosen using the ERR SCS PIN field. Disable the $\overline{\text{SPICS}}$ error-inducing logic. Enable the $\overline{\text{SPICS}}$ error-inducing logic.
1	LPBK TYPE	0 1	Module I/O loopback type (analog/digital). See <a href="#">Figure 12-22</a> for the different types of loopback modes. Enable Digital loopback when IOLPBKSTENA = 1010. Enable Analog loopback when IOLPBKSTENA = 1010.
0	RXPENA	0 1	Enable analog loopback through the receive pin. <b>Note: This bit is valid only when LPBK TYPE = 1, which chooses analog loopback mode.</b> Analog loopback is through the transmit pin. Analog loopback is through the receive pin.

### 12.8.45 SPI Extended Prescale Register 1 (EXTENDED\_PRESCALE1 for SPIFMT0 and SPIFMT1)

This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This is an extension of SPIFMT0 and SPIFMT1 registers. For example, EPRESCALE\_FMT1[7:0] of EXTENDED\_PRESCALE1 and PRESCALE1 of SPIFMT1 register will always reflect the same contents. Similarly, EPRESCALE\_FMT0[7:0] and PRESCALE0 of SPIFMT0 reflect the same contents.

**Figure 12-74. SPI Extended Prescale Register 1 (EXTENDED\_PRESCALE1 for SPIFMT0 and SPIFMT1)**  
[offset = 138h]

31	27	26	16
Reserved		EPRESCALE_FMT1	
R-0		R/WP-0	
15	11	10	0
Reserved		EPRESCALE_FMT0	
R-0		R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-52. SPI Extended Prescale Register 1 (EXTENDED\_PRESCALE1) Field Descriptions**

Bit	Field	Value	Description
31-27	Reserved	0	Reads return 0. Writes have no effect.
26-16	EPRESCALE_FMT1	0-7FFh	<p>EPRESCALE_FMT1. Extended Prescale value for SPIFMT1. EPRESCALE_FMT1 determines the bit transfer rate of data format 1 if the SPI/MibSPI is the network master. EPRESCALE_FMT1 is use to derive SPICLK from VCLK. If the SPI is configured as slave, EPRESCALE_FMT1 <b>does not need</b> to be configured. These EPRESCALE_FMT1[7:0] bits and PRESCALE1 bits of SPIFMT1 register will point to the same physically implemented register. The clock rate for data format 1 can be calculated as:</p> $BR_{Format1} = VCLK / (EPRESCALE\_FMT1 + 1)$ <p>Write: This register field should be written if a SPICLK prescaler of more VCLK/256 is required. This field provides a prescaler of up to VCLK/2048 for SPICLK. Writing to this register field will also get reflected in the PRESCALE1 bits of SPIFMT1 register.</p> <p>Read: Reading this field will reflect the PRESCALE value based on the last written register field, that is, EXTENDED_PRESCALE1[26:16] or SPIFMT1[15:8] register.</p> <p><b>Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE1 register is programmed after SPIFMT1 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE1 register when a prescale of more 256 is intended on SPICLK. Writing to PRESCALE1 field of SPIFMT1 will automatically clear EPRESCALE_FMT1[10:8] bits to 000 so that the integrity of PRESCALE value is maintained.</b></p>
15-11	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-52. SPI Extended Prescale Register 1 (EXTENDED\_PRESCALE1) Field Descriptions (continued)**

Bit	Field	Value	Description
10-0	EPRESCALE_FMT0	0-7FFh	<p>EPRESCALE_FMT0. Extended Prescale value for SPIFMT0. EPRESCALE_FMT0 determines the bit transfer rate of data format 0 if the SPI/MibSPI is the network master. EPRESCALE_FMT0 is use to derive SPICLK from VCLK. If the SPI is configured as slave, EPRESCALE_FMT0 <b>does not need</b> to be configured. These EPRESCALE_FMT0[7:0] bits and PRESCALE0 bits of SPIFMT0 register will point to the same physically implemented register. The clock rate for data format 0 can be calculated as:</p> $BR_{Format0} = VCLK / (EPRESCALE\_FMT0 + 1)$ <p>Write: This register field should be written if a SPICLK prescaler of more VCLK/256 is required. This field provides a prescaler of up to VCLK/2048 for SPICLK. Writing to this register field will also get reflected in the PRESCALE0 bits of SPIFMT0 register.</p> <p>Read: Reading this field will reflect the PRESCALE value based on the last written register field, that is, EXTENDED_PRESCALE0[10:0] or SPIFMT0[15:8] register.</p> <p><b>Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE1 register is programmed after SPIFMT0 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE1 register when a prescale of more 256 is intended on SPICLK. Writing to PRESCALE0 field of SPIFMT0 will automatically clear EPRESCALE_FMT0[10:8] bits to 000 so that the integrity of PRESCALE value is maintained.</b></p>

### 12.8.46 SPI Extended Prescale Register 2 (EXTENDED\_PRESCALE2 for SPIFMT2 and SPIFMT3)

This register provides an extended Prescale values for SPICLK generation to be able to interface with much slower SPI Slaves. This is an extension of SPIFMT2 and SPIFMT3 registers. For example, EPRESCALE\_FMT3[7:0] of EXTENDED\_PRESCALE2 and PRESCALE3 of SPIFMT3 register will always reflect the same contents. Similarly, EPRESCALE\_FMT2[7:0] and PRESCALE2 of SPIFMT2 reflect the same contents.

**Figure 12-75. SPI Extended Prescale Register 2 (EXTENDED\_PRESCALE2 for SPIFMT2 and SPIFMT3)**  
[offset = 13Ch]

31	27	26	16
Reserved		EPRESCALE_FMT3	
R-0		R/WP-0	
15	11	10	0
Reserved		EPRESCALE_FMT2	
R-0		R/WP-0	

LEGEND: R/W = Read/Write; R = Read only; WP = Write in privilege mode only; -n = value after reset

**Table 12-53. SPI Extended Prescale Register 2 (EXTENDED\_PRESCALE2) Field Descriptions**

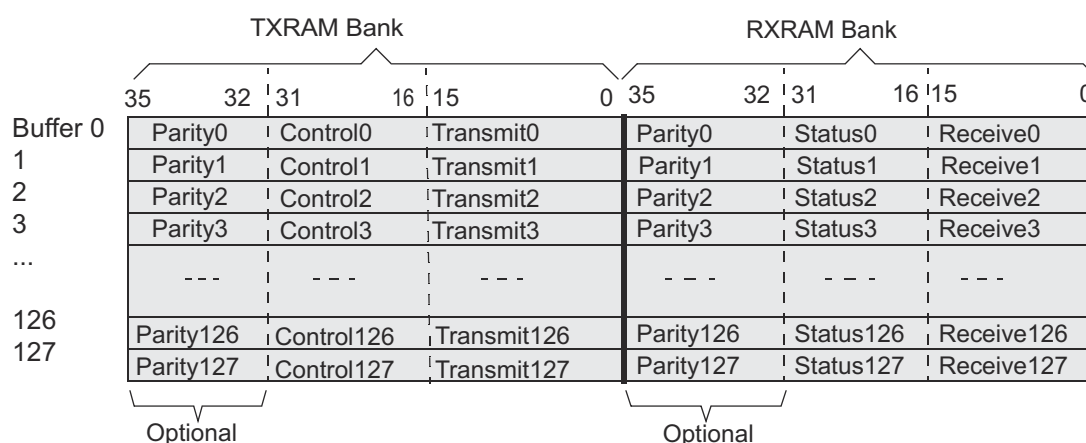
Bit	Field	Value	Description
31-27	Reserved	0	Reads return 0. Writes have no effect.
26-16	EPRESCALE_FMT3	0-7FFh	<p>EPRESCALE_FMT3. Extended Prescale value for SPIFMT3. EPRESCALE_FMT3 determines the bit transfer rate of data format 3 if the SPI/MibSPI is the network master. EPRESCALE_FMT3 is use to derive SPICLK from VCLK. If the SPI is configured as slave, EPRESCALE_FMT3 <b>does not need</b> to be configured. These EPRESCALE_FMT3[7:0] bits and PRESCALE3 bits of SPIFMT3 register will point to the same physically implemented register. The clock rate for data format 1 can be calculated as:</p> $BR_{Format3} = VCLK / (EPRESCALE\_FMT3 + 1)$ <p>Write: This register field should be written if a SPICLK prescaler of more VCLK/256 is required. This field provides a prescaler of up to VCLK/2048 for SPICLK. Writing to this register field will also get reflected in the PRESCALE3 bits of SPIFMT3 register.</p> <p>Read: Reading this field will reflect the PRESCALE value based on the last written register field, that is, EXTENDED_PRESCALE3[26:16] or SPIFMT3[15:8] register.</p> <p><b>Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE2 register is programmed after SPIFMT3 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE2 register when a prescale of more 256 is intended on SPICLK. Writing to PRESCALE3 field of SPIFMT3 will automatically clear EPRESCALE_FMT3[10:8] bits to 000 so that the integrity of PRESCALE value is maintained.</b></p>
15-11	Reserved	0	Reads return 0. Writes have no effect.

**Table 12-53. SPI Extended Prescale Register 2 (EXTENDED\_PRESCALE2) Field Descriptions (continued)**

Bit	Field	Value	Description
10-0	EPRESCALE_FMT2	0-7FFh	<p>EPRESCALE_FMT2. Extended Prescale value for SPIFMT2. EPRESCALE_FMT2 determines the bit transfer rate of data format 2 if the SPI/MibSPI is the network master. EPRESCALE_FMT2 is use to derive SPICLK from VCLK. If the SPI is configured as slave, EPRESCALE_FMT2 <b>does not need</b> to be configured. These EPRESCALE_FMT2[7:0] bits and PRESCALE2 bits of SPIFMT2 register will point to the same physically implemented register. The clock rate for data format 0 can be calculated as:</p> $BR_{Format2} = VCLK / (EPRESCALE_FMT2 + 1)$ <p>Write: This register field should be written if a SPICLK prescaler of more VCLK/256 is required. This field provides a prescaler of up to VCLK/2048 for SPICLK. Writing to this register field will also get reflected in the PRESCALE2 bits of SPIFMT2 register.</p> <p>Read: Reading this field will reflect the PRESCALE value based on the last written register field, that is, EXTENDED_PRESCALE2[10:0] or SPIFMT2[15:8] register.</p> <p><b>Note: If Extended Prescaler is required, it should be ensured that EXTENDED_PRESCALE2 register is programmed after SPIFMT2 register is programmed. This is to ensure that the final SPICLK prescale value is controlled by EXTENDED_PRESCALE2 register when a prescale of more 256 is intended on SPICLK. Writing to PRESCALE2 field of SPIFMT2 will automatically clear EPRESCALE_FMT2[10:8] bits to 000 so that the integrity of PRESCALE value is maintained.</b></p>

## 12.9 Multi-Buffer RAM

The multi-buffer RAM is used for holding transmit and received data, control and status information. The multi-buffer RAM contains two banks of up to 128 32-bit words for a maximum configuration. One bank (TXRAM) contains entries for transmit data (replicating the SPIDAT1 register). The other bank (RXRAM) contains received data (replicating the SPIBUF register). The buffers can be partitioned into multiple TGs, each containing a programmable number of buffers. Each of the buffers can be subdivided into 16-bit transmit field, 16-bit receive field, 16-bit control field, and 16-bit status field, as displayed in Figure 12-76. A 4-bit parity field per word is also included in each bank of RAM.



**Figure 12-76. Multi-Buffer RAM Configuration**

All fields can be read and written with 8-bit, 16-bit, or 32-bit accesses.

The transmit fields can be written and read in the address range 000h to 1FFh. The transmit words contain data and control fields.

The receive RAM fields are read-only and can be accessed through the address range 200h to 3FCh. The receive words contain data and status fields.

The chip select number (CSNR) bit field of the control field for a given word is mirrored into the corresponding receive-buffer status field after transmission.

The Parity is automatically calculated and copied to Parity location

### Note

Please refer to the specific device datasheet for the actual number of transmit and receive buffers.

Write to unimplemented buffer is overwriting the corresponding implemented buffer. In MIBSPI, if the RAM SIZE specified is 32 buffers, write to 33rd buffer overwrites 1st buffer, write to 34th buffer overwrites 2nd buffer and so on.



### 12.9.1 Multi-Buffer RAM Auto Initialization

When the MIBSPI is out of reset mode, auto initialization of multi-buffer RAM starts. The application code must check for BUFINITACTIVE bit to be 0 (Multi-buffer RAM initialization is complete) before configuring multi-buffer RAM.

Besides the default auto initialization after reset, the auto-initialization sequence can also be done by:

1. Enable the global hardware memory initialization key by programming a value of 1010b to the bits [3:0] of the MINITGCR register of the System module.
2. Set the control bit for the multi-buffer RAM in the MSINENA System module register. This bit is device-specific for each memory that support auto-initialization. Please refer to the device datasheet to identify the control bit for the multi-buffer RAM. This starts the initialization process. The BUFINITACTIVE bit will get set to reflect that the initialization is ongoing.
3. When the memory initialization is completed, the corresponding status bit in the MINISTAT register will be set. Also, the BUFINITACTIVE bit will get cleared.
4. Disable the global hardware memory initialization key by programming a value of 0101 to the bits [3:0] of the MINITGCR register of the System module.

Please refer to the *Architecture* chapter for more details on the memory auto-initialization process.

#### Note

During Auto Initialization process, all the Multi-buffer mode registers (except MIBSPIE) will be reset to their default values. So, it should be ensured that Auto Initialization is completed before configuring the Multi-buffer mode registers.

### 12.9.2 Multi-Buffer RAM Register Summary

This section describes the multi-buffer RAM control and transmit-data fields of each word of TXRAM, and the status and receive-data fields of each word of RXRAM. The base address for multi-buffer RAM is FF0E 0000h for MibSPI1 RAM, FF0C 000h for MibSPI3 RAM, and FF0A 0000h for MibSPI5 RAM.

**Table 12-54. Multi-Buffer RAM Register Summary**

Offset	Acronym	Register Description	Section
Base + 0h-1FFh	TXRAM	Multi-Buffer RAM Transmit Data Register	<a href="#">Section 12.9.3</a>
Base + 200h-3FFh	RXRAM	Multi-Buffer RAM Receive Buffer Register	<a href="#">Section 12.9.4</a>

### 12.9.3 Multi-Buffer RAM Transmit Data Register (TXRAM)

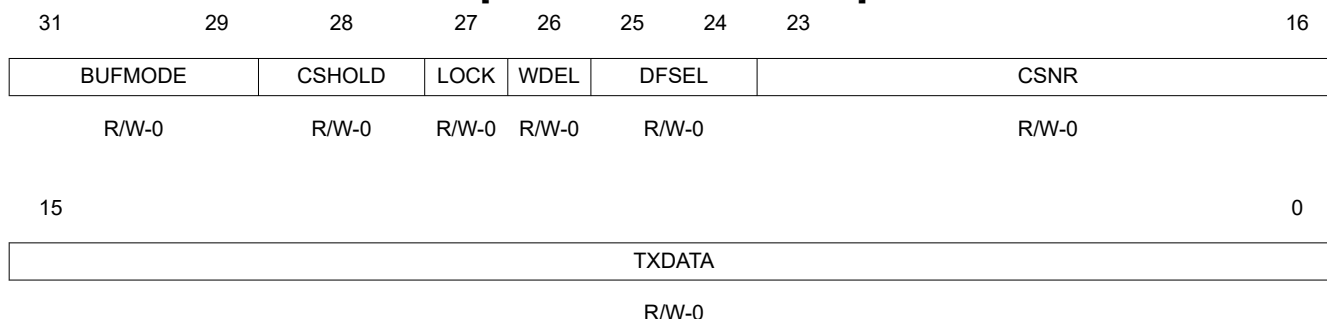
Each word of TXRAM is a transmit-buffer register.

#### Note

Writing to only the control fields, bits 28 through 16, does not initiate any SPI transfer in master mode. This feature can be used to set up SPICLK phase or polarity before actually starting the transfer by only updating the DFSEL bit field to select the required phase and polarity combination.

**Figure 12-77. Multi-Buffer RAM Transmit Data Register (TXRAM)**

[offset = RAM Base + 0h-1FFh]



LEGEND: R/W = Read/Write; -n = value after reset

**Table 12-55. Multi-Buffer RAM Transmit Data Register (TXRAM) Field Descriptions**

Bit	Field	Value	Description
31-29	BUFMODE		Specify conditions that are recognized by the sequencer to initiate transfers of each buffer word. When one of the "skip" modes is selected, the sequencer checks the buffer status every time it reads from this buffer. If the current buffer status (TXFULL, RXEMPTY) does not match, the buffer is skipped without a data transfer. When one of the "suspend" modes is selected, the sequencer checks the buffer status when it reads from this buffer. If TxFULL and/or RXEMPTY do not match, the sequencer waits until a match occurs. No data transfer is initiated until the status condition of this buffer changes.
		0	<b>disabled.</b> The buffer is disabled.
		1h	<b>skip single-transfer mode.</b> Skip this buffer until the corresponding TxFULL flag is set (new transmit data is available).
		2h	<b>skip overwrite-protect mode.</b> Skip this buffer until the corresponding RXEMPTY flag is set (new receive data can be stored in RXDATA without data loss).
		3h	<b>skip single-transfer overwrite-protect mode.</b> Skip this buffer until both of the corresponding TxFULL and RXEMPTY flags are set. (new transmit data available and previous data received by the host).
		4h	<b>continuous mode.</b> Initiate a transfer each time the sequencer checks this buffer. Data words are retransmitted if the buffer has not been updated. Receive data is overwritten, even if it has not been read.
		5h	<b>suspend single-transfer mode.</b> Suspend-to-wait until the corresponding TxFULL flag is set (the sequencer stops at the current buffer until new transmit data is written in the TXDATA field).
		6h	<b>suspend overwrite-protect mode.</b> Suspend-to-wait until the corresponding RXEMPTY flag is set (the sequencer stops at the current buffer until the previously-received data is read by the host).
		7h	<b>suspend single-transfer overwrite-protect mode.</b> Suspend-to-wait until the corresponding TxFULL and RXEMPTY flags are set (the sequencer stops at the current buffer until new transmit data is written into the TXDATA field and the previously-received data is read by the host).

**Table 12-55. Multi-Buffer RAM Transmit Data Register (TXRAM) Field Descriptions (continued)**

Bit	Field	Value	Description
28	CSHOLD	0 1	<p>Chip select hold mode. The CSHOLD bit is supported in master mode only, it is ignored in slave mode. CSHOLD defines the behavior of the chip select line at the end of a data transfer.</p> <p>The chip select signal is deactivated at the end of a transfer after the T2CDELAY time has passed. If two consecutive transfers are dedicated to the same chip select this chip select signal will be deactivated for at least 2VCLK cycles before it is activated again.</p> <p>The chip select signal is held active at the end of a transfer until a control field with new data and control information is loaded into SPIDAT1. If the new chip select number equals the previous one, the active chip select signal is extended until the end of transfer with CSHOLD cleared, or until the chip-select number changes.</p>
27	LOCK	0 1	<p>Lock two consecutive buffer words. Do not allow interruption by TGs with higher priority.</p> <p>Any higher-priority TG can begin at the end of the current transaction.</p> <p>A higher-priority TG cannot occur until after the next unlocked buffer word is transferred.</p>
26	WDEL	0 1	<p>Enable the delay counter at the end of the current transaction.</p> <p><b>Note: The WDEL bit is supported in master mode only. In slave mode, this bit is ignored.</b></p> <p>No delay will be inserted. However, <math>\overline{\text{SPICS}}</math> pins will still be de-activated for at least for 2VCLK cycles if CSHOLD = 0.</p> <p><b>Note: The duration for which the <math>\overline{\text{SPICS}}</math> pin remains deactivated also depends upon the time taken to supply a new word after completing the shift operation (in compatibility mode). If TXBUF is already full, then the <math>\overline{\text{SPICS}}</math> pin will be deasserted for at least two VCLK cycles (if WDEL = 0).</b></p> <p>After a transaction, WDELAY of the corresponding data format will be loaded into the delay counter. No transaction will be performed until the WDELAY counter overflows. The SPICS pins will be de-activated for at least (WDELAY + 2) × VCLK_Period duration.</p>
25-24	DFSEL	0 1h 2h 3h	<p>Data word format select.</p> <p>Data word format 0 is selected.</p> <p>Data word format 1 is selected.</p> <p>Data word format 2 is selected.</p> <p>Data word format 3 is selected.</p>
23-16	CSNR	0-FFh	<p>Chip select (CS) number. CSNR defines the chip select pins that will be activated during the data transfer. CSNR is a bit-mask that controls all chip select pins. See <a href="#">Table 12-56</a>.</p> <p><b>Note: If your MibSPI has less than 8 chip select pins, all unused upper bits will be 0. For example, MiBSPI3 has 6 chip select pins, if you write FFh to CSNR, the actual number stored in CSNR is 3Fh.</b></p>
15-0	TXDATA	0-7FFFh	<p>Transfer data. When written, these bits are copied to the shift register if it is empty. If the shift register is not empty, then they are held in TXBUF.</p> <p>SPIEN must be set to 1 before this register can be written to. Writing a 0 to SPIEN forces the lower 16 bits of TXDATA to 0.</p> <p>A write to this register (or to the TXDATA field only) drives the contents of the CSNR field on the <math>\overline{\text{SPICS}}</math> pins, if the pins are configured as functional pins (automatic chip select, see <a href="#">Section 12.2</a>).</p> <p>When this register is read, the contents of TXBUF, which holds the latest data written, will be returned.</p> <p><b>Note: Regardless of the character length, the transmit data should be right-justified before writing to the SPIDAT1 register.</b></p>

**Table 12-56. Chip Select Number Active**

CSNR Value	Chip Select Active:						CSNR Value	Chip Select Active:					
	CS[5] <sup>(1)</sup>	CS[4] <sup>(1)</sup>	CS[3] <sup>(1)</sup>	CS[2] <sup>(1)</sup>	CS[1] <sup>(1)</sup>	CS[0]		CS[5] <sup>(1)</sup>	CS[4] <sup>(1)</sup>	CS[3] <sup>(1)</sup>	CS[2] <sup>(1)</sup>	CS[1] <sup>(1)</sup>	CS[0]
0h	No chip select pin is active.						20h	x					
1h						x	21h	x					x
2h					x		22h	x				x	
3h					x	x	23h	x				x	x
4h				x			24h	x			x		
5h				x		x	25h	x			x		x
6h				x	x		26h	x			x	x	
7h				x	x	x	27h	x			x	x	x
8h			x				28h	x		x			
9h			x			x	29h	x		x			x
Ah			x		x		2Ah	x		x		x	
Bh			x		x	x	2Bh	x		x		x	x
Ch			x	x			2Ch	x		x	x		
Dh			x	x		x	2Dh	x		x	x		x
Eh			x	x	x		2Eh	x		x	x	x	
Fh			x	x	x	x	2Fh	x		x	x	x	x
10h		x					30h	x	x				
11h		x				x	31h	x	x				x
12h		x			x		32h	x	x			x	
13h		x			x	x	33h	x	x			x	x
14h		x		x			34h	x	x		x		
15h		x		x		x	35h	x	x		x		x
16h		x		x	x		36h	x	x		x	x	
17h		x		x	x	x	37h	x	x		x	x	x
18h		x	x				38h	x	x	x			
19h		x	x			x	39h	x	x	x			x
1Ah		x	x		x		3Ah	x	x	x		x	
1Bh		x	x		x	x	3Bh	x	x	x		x	x
1Ch		x	x	x			3Ch	x	x	x	x		
1Dh		x	x	x		x	3Dh	x	x	x	x		x
1Eh		x	x	x	x		3Eh	x	x	x	x	x	
1Fh		x	x	x	x	x	3Fh	x	x	x	x	x	x

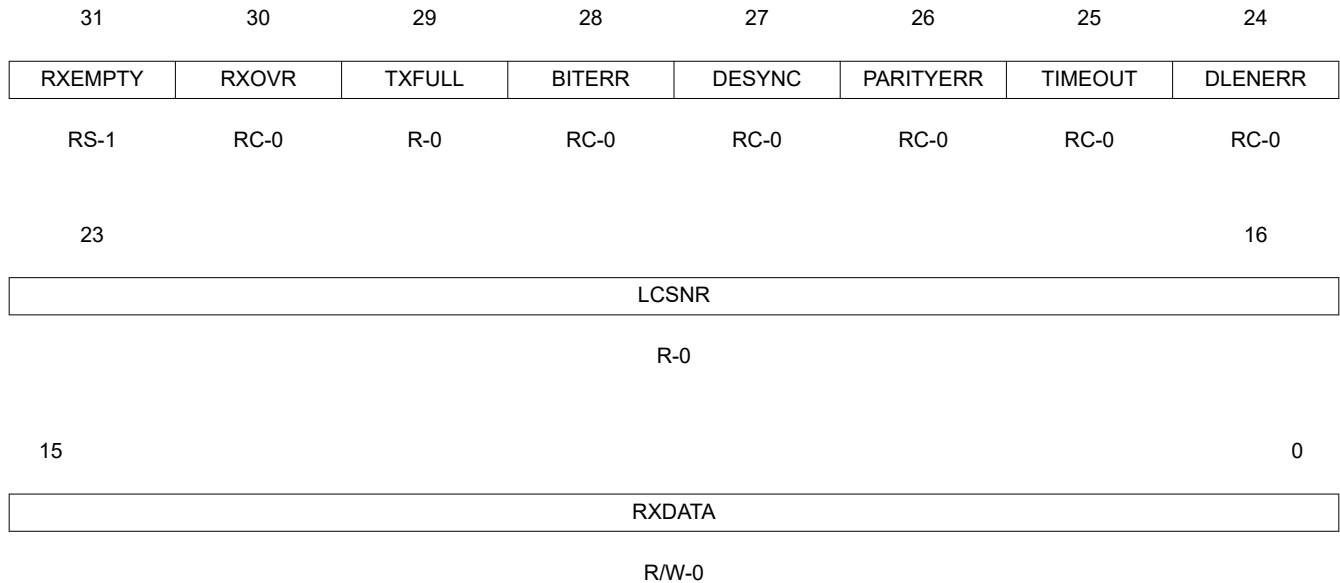
(1) If your MibSPI does not have this chip select pin, this bit is 0.

### 12.9.4 Multi-Buffer RAM Receive Buffer Register (RXRAM)

Each word of RXRAM is a receive-buffer register.

**Figure 12-78. Multi-Buffer RAM Receive Buffer Register (RXRAM)**

**[offset = RAM Base + 200h-3FFh]**



LEGEND: R/W = Read/Write; R = Read only; C = Clear; S = Set; -n = value after reset

**Table 12-57. Multi-Buffer Receive Buffer Register (RXRAM) Field Descriptions**

Bit	Field	Value	Description
31	RXEMPTY	<div>0</div> <div>1</div>	<p>Receive data buffer empty. When the host reads the RXDATA field or the entire RXRAM register, it automatically sets the RXEMPTY flag. When a data transfer is completed, the received data is copied into RXDATA, and the RXEMPTY flag is cleared.</p> <p>New data has been received and copied into RXDATA.</p> <p>No data has been received since the last read of RXDATA.</p> <p>This flag gets set to 1 under the following conditions:</p> <ul style="list-style-type: none"> <li>Reading the RXDATA field of the RXRAM register</li> <li>Writing a 1 to clear the RXINTFLG bit in the SPI Flag Register (SPIFLG)</li> </ul> <p>Write-clearing the RXINTFLG bit before reading RXDATA indicates the received data is being ignored. Conversely, RXINTFLG can be cleared by reading the RXDATA field of RXRAM (or the entire register).</p>
30	RXOVR	<div>0</div> <div>1</div>	<p>Receive data buffer overrun. When a data transfer is completed and the received data is copied into RXBUF while it is already full, RXOVR is set. Overruns always occur to RXBUF, not to RXRAM; the contents of RXRAM are overwritten only after it is read by the Peripheral (VBUSP) master (CPU, DMA, or other host processor).</p> <p>If enabled, the RXOVRN interrupt is generated when RXBUF is overwritten, and reading either SPI Flag Register (SPIFLG) or SPIVEXTx shows the RXOVRN condition. Two read operations from the RXRAM register are required to reach the overwritten buffer word (one to read RXRAM, which then transfers RXDATA into RXRAM for the second read).</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the RXRAM register is read.</b></p> <p><b>Note: A special condition under which RXOVR flag gets set. If both RXRAM and RXBUF are already full and while another buffer receive is underway, if any errors such as TIMEOUT, BITERR, and DLEN_ERR occur, then RXOVR in RXBUF and SPI Flag Register (SPIFLG) registers will be set to indicate that the status flags are getting overwritten by the new transfer. This overrun should be treated like a normal receive overrun.</b></p> <p>No receive data overrun condition occurred since last read of the data field.</p> <p>A receive data overrun condition occurred since last read of the data field.</p>

**Table 12-57. Multi-Buffer Receive Buffer Register (RXRAM) Field Descriptions (continued)**

Bit	Field	Value	Description
29	TXFULL	0 1	<p>Transmit data buffer full. This flag is a read-only flag. Writing into the SPIDAT0 or SPIDAT1 field while the TX shift register is full will automatically set the TXFULL flag. Once the word is copied to the shift register, the TXFULL flag will be cleared. Writing to SPIDAT0 or SPIDAT1 when both TXBUF and the TX shift register are empty does not set the TXFULL flag.</p> <p>The transmit buffer is empty; SPIDAT0/SPIDAT1 is ready to accept a new data.</p> <p>The transmit buffer is full; SPIDAT0/SPIDAT1 is not ready to accept new data.</p>
28	BITERR	0 1	<p>Bit error. There was a mismatch of internal transmit data and transmitted data.</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the RXRAM register is read.</b></p> <p>No bit error occurred.</p> <p>A bit error occurred. The SPI samples the signal of the transmit pins (master: SIMOx, slave: SOMIx) at the receive point (one-half clock cycle after the transmit point). If the sampled value differs from the transmitted value, a bit error is detected and the BITERR flag is set. Possible reasons for a bit error include noise, an excessively high bit rate, capacitive load, or another master/slave trying to transmit at the same time.</p>
27	DESYNC	0 1	<p>Desynchronization of slave device. This bit is valid in master mode only.</p> <p>The master monitors the ENA signal coming from the slave device and sets the DESYNC flag if ENA is deactivated before the last reception point or after the last bit is transmitted plus <math>t_{T2DELAY}</math>. If DESYNCENA is set, an interrupt is asserted. Desynchronization can occur if a slave device misses a clock edge coming from the master.</p> <p><b>Note: In the Compatibility Mode MibSPI, under some circumstances it is possible for a desync error detected for the previous buffer to be visible in the current buffer. This is because the receive completion flag/interrupt is generated when the buffer transfer is completed. But desynchronization is detected after the buffer transfer is completed. So, if the VBUS master reads the received data quickly when an RXINT is detected, then the status flag may not reflect the correct desync condition. In multi-buffer mode, the desync flag is always guaranteed to be for the current buffer.</b></p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the RXRAM register is read.</b></p> <p>No slave desynchronization is detected.</p> <p>A slave device is desynchronized.</p>
26	PARITYERR	0 1	<p>Parity error. The calculated parity differs from the received parity bit.</p> <p>If the parity generator is enabled (selected individually for each buffer) an even or odd parity bit is added at the end of a data word. During reception of the data word, the parity generator calculates the reference parity and compares it to the received parity bit. If a mismatch is detected, the PARITYERR flag is set.</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the RXRAM register is read.</b></p> <p>No parity error is detected.</p> <p>A parity error occurred.</p>
25	TIMEOUT	0 1	<p>Time-out because of non-activation of <math>\overline{\text{SPIEN}}_A</math> pin.</p> <p>The SPI generates a time-out when the slave does not respond in time by activating the ENA signal after the chip select signal has been activated. If a time-out condition is detected, the corresponding chip select is deactivated immediately and the TIMEOUT flag is set. In addition, the TIMEOUT flag in the status field of the corresponding buffer and in the SPI Flag Register (SPIFLG) is set.</p> <p><b>Note: This bit is valid only in master mode.</b></p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the RXRAM register is read.</b></p> <p>No <math>\overline{\text{SPIEN}}_A</math> pin time-out occurred.</p> <p>An <math>\overline{\text{SPIEN}}_A</math> signal time-out occurred.</p>
24	DLENERR	0 1	<p>Data length error flag.</p> <p><b>Note: This flag is cleared to 0 when the RXDATA field of the RXRAM register is read.</b></p> <p>No data-length error occurred.</p> <p>A data length error occurred.</p>
23-16	LCSNR	0-FFh	<p>Last chip select number. LCSNR in the status field is a copy of CSNR in the corresponding control field. It contains the chip select number that was activated during the last word transfer.</p>

**Table 12-57. Multi-Buffer Receive Buffer Register (RXRAM) Field Descriptions (continued)**

Bit	Field	Value	Description
15-0	RXDATA	0-FFFFh	SPI receive data. This is the received word, transferred from the receive shift-register at the end of a transfer. Regardless of the programmed character length and the direction of shifting, the received data is stored right-justified in the register.

## 12.10 Parity Memory

The parity portion of multi-buffer RAM is not accessible by the CPU during normal operating modes. However, each read or write operation to the control/data/status portion of the multi-buffer RAM causes reads/writes to the parity portion as well.

- Each write to the multi-buffer RAM (either from the Peripheral interface or by the MibSPI itself) causes a write operation to the parity portion of RAM simultaneously to update the equivalent parity bits.
- Each read operation from the multi-buffer RAM (either from the Peripheral interface or by the MibSPI itself) causes a read operation from the parity portion of the RAM for parity comparison purpose.
- Reads/Writes to multi-buffer RAM can either be caused by any CPU/DMA accesses or by the sequencer logic of MibSPI itself.
- In case of Parity error ESM module is notified to generate MIBSPI Parity ESM interrupt. User can check the error status and address location captured in the UERRSTAT and UERRADDRx registers respectively.

For testing the parity portion of the multi-buffer RAM, which is a 4-bit field per word address (1 bit per byte), a separate parity memory test mode is available. Parity memory test mode can be enabled and disabled by the PTESTEN bit in the UERRCTRL register.

During the parity test mode, the parity locations are addressable at the address between RAM\_BASE\_ADDR + 0x400h and RAM\_BASE\_ADDR + 0x7FFh. Each location corresponds, sequentially, to each TXRAM word, then to each RXRAM word. See [Figure 12-79](#) for a diagram of the memory map of parity memory during normal operating mode and during parity test mode.

During parity test mode, after writing the data/control portion of the RAM, the parity locations can be written with incorrect parity bits to intentionally cause parity errors.

See the device-specific data sheet to get the actual base address of the multi-buffer RAM.

---

### Note

The RX\_RAM\_ACCESS bit can also be set to 1 during the parity test mode to be enable writes to RXRAM locations. Both parity RAM testing and RXRAM testing can be done together.

---

There are 4 bits of parity corresponding to each of the 32-bit multi-buffer locations. Individual bits in the parity memory are byte-addressable in parity test mode. See the example in [Figure 12-80](#) for further details.

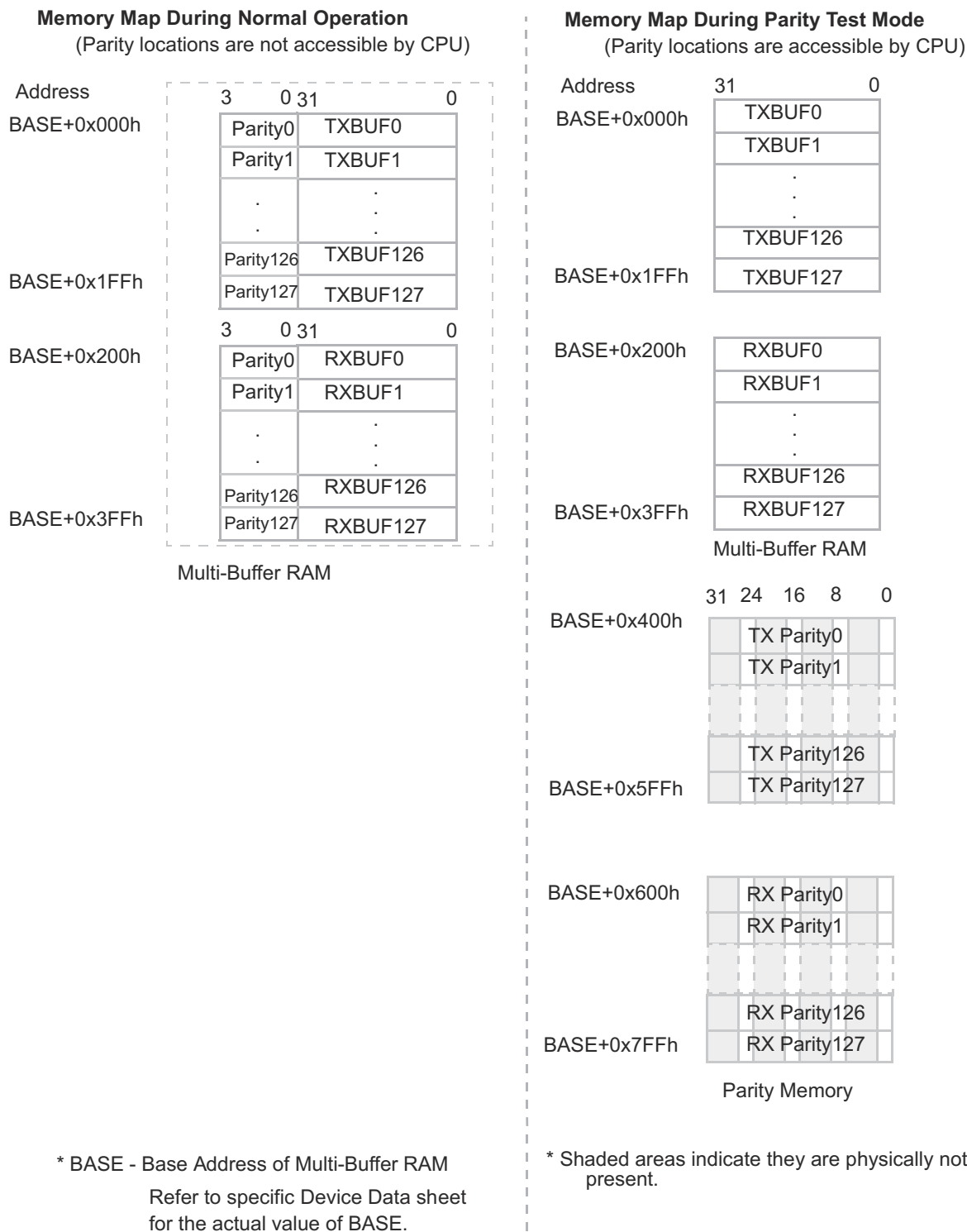
---

### Note

Polarity of the parity (odd/even) varies by device. In some devices, a control register in the system module can be used to select odd or even parity.

---



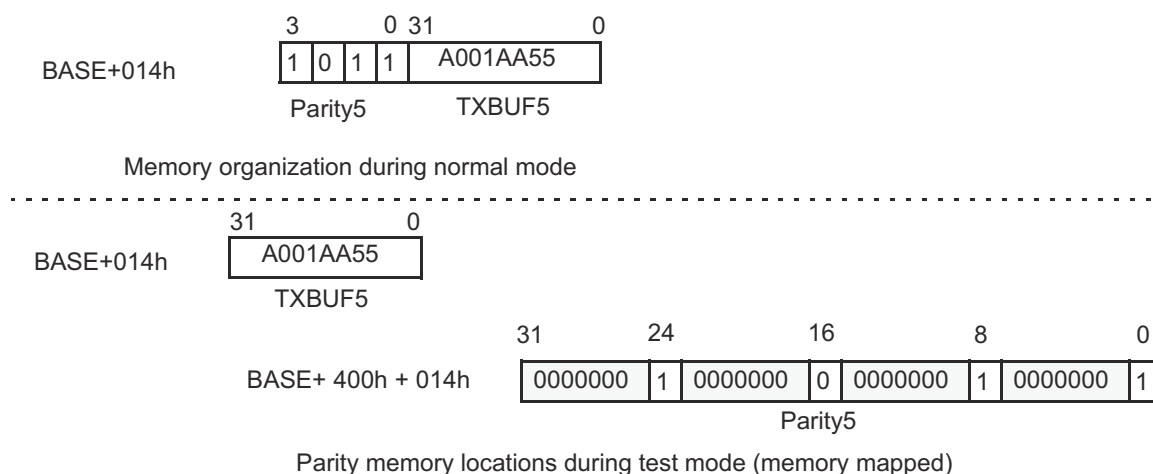


**Figure 12-79. Memory Map for Parity Locations During Normal and Test Mode**

### 12.10.1 Example of Parity Memory Organization

Suppose TXBUF5 (6th location in TXRAM) in the multi-buffer RAM is written with a value of A001\_AA55. If the polarity of the parity is set to odd, the corresponding parity location parity5 will get updated with equivalent parity of 1011 in its field.

During parity-memory test mode, these bits can be individually byte addressed. The return data will be a byte adjusted with actual parity bit in the LSB of the byte. If a word is read from the word-boundary address of parity locations, then each bit of the 4-bit parity is byte-adjusted and a 32-bit word is returned. 0s will be padded into the parity bits to get each byte. See Figure 12-80 for a diagram.



- 1 Shaded areas indicate reads return 0, writes have no effect. These registers are not physically present.

**Figure 12-80. Example of Memory-Mapped Parity Locations During Test Mode**

#### Note

##### Read Access to Parity Memory Locations

Parity memory locations can be read even without entering into parity memory test mode. Their address remains as in memory test mode. It is only to enter parity-memory test mode to enable write access to the parity memory locations.

## 12.11 MibSPI Pin Timing Parameters

The pin timings of SPI can be classified based on its mode of operation. In each mode, different configurations like Phase and Polarity affect the pin timings.

The pin directions are based on the mode of operation.

### Controller mode SPI:

- SPICLK (SPI Clock) - Output
- SPISIMO (SPI Peripheral In Controller Out) - Output
- $\overline{\text{SPICS}}$  (SPI Peripheral Chip Selects) - Output
- SPISOMI (SPI Peripheral Out Controller In) - Input
- $\overline{\text{SPIENA}}$  (SPI Peripheral ready Enable) - Input

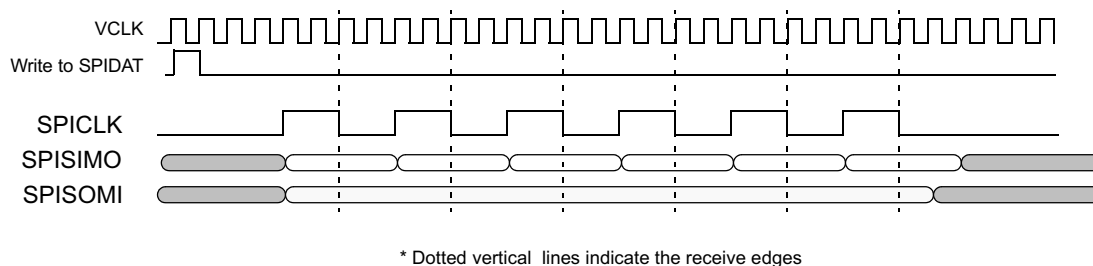
### Peripheral mode SPI:

- SPICLK - Input
- SPISIMO - Input
- $\overline{\text{SPICS}}$  - Input
- SPISOMI - Output
- $\overline{\text{SPIENA}}$  - Output

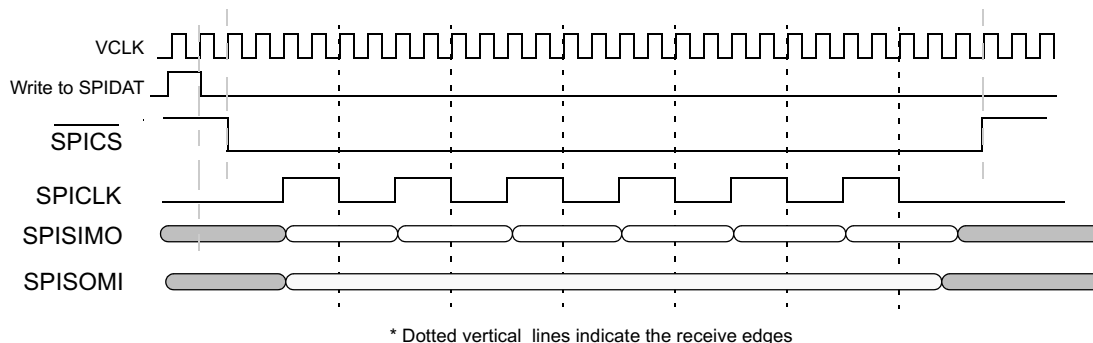
### Note

All the timing diagrams given below are with Phase = 0 and Polarity = 0, unless explicitly stated otherwise.

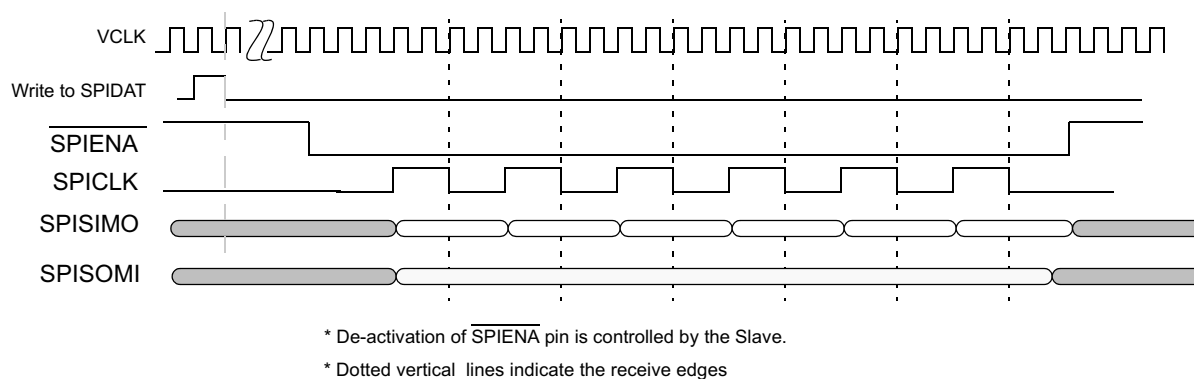
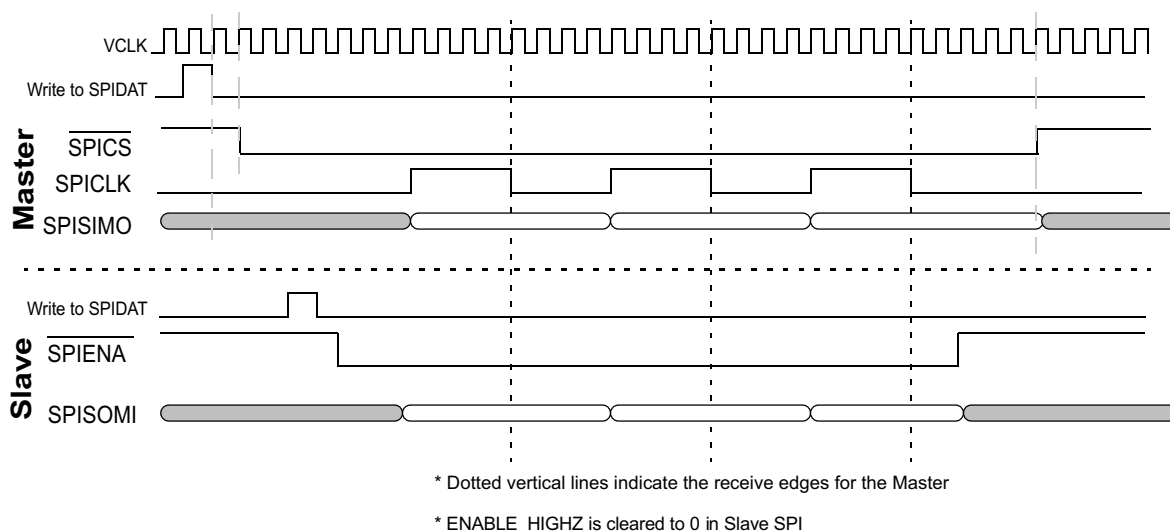
#### 12.11.1 Controller Mode Timings for SPI/MibSPI



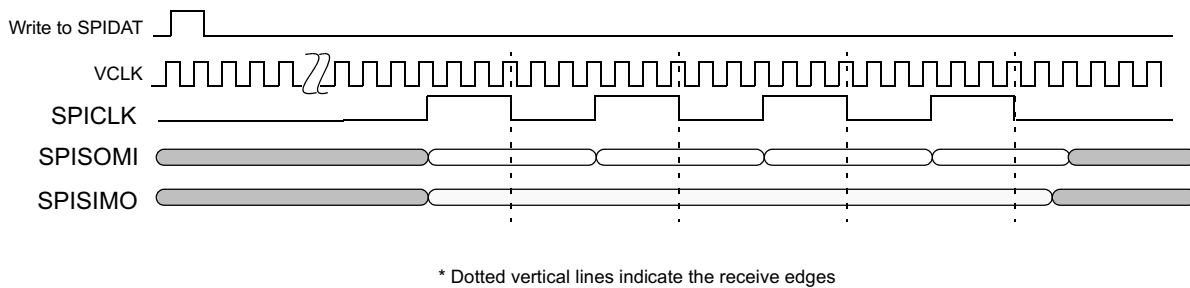
**Figure 12-81. SPI/MibSPI Pins During Controller Mode 3-pin Configuration**



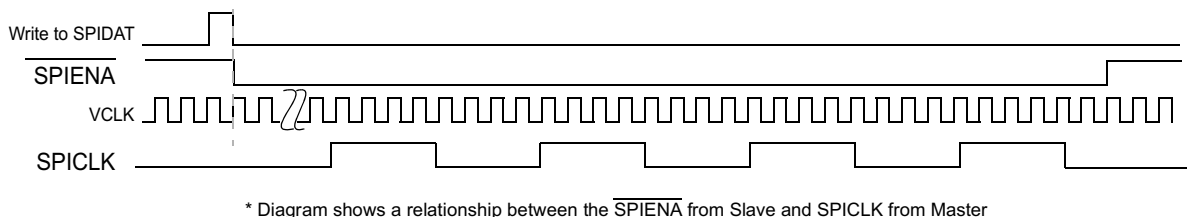
**Figure 12-82. SPI/MibSPI Pins During Controller Mode 4-pin with  $\overline{\text{SPICS}}$  Configuration**


**Figure 12-83. SPI/MibSPI Pins During Controller Mode 4-pin with  $\overline{\text{SPIENA}}$  Configuration**

**Figure 12-84. SPI/MibSPI Pins During Controller/Peripheral Mode with 5-pin Configuration**

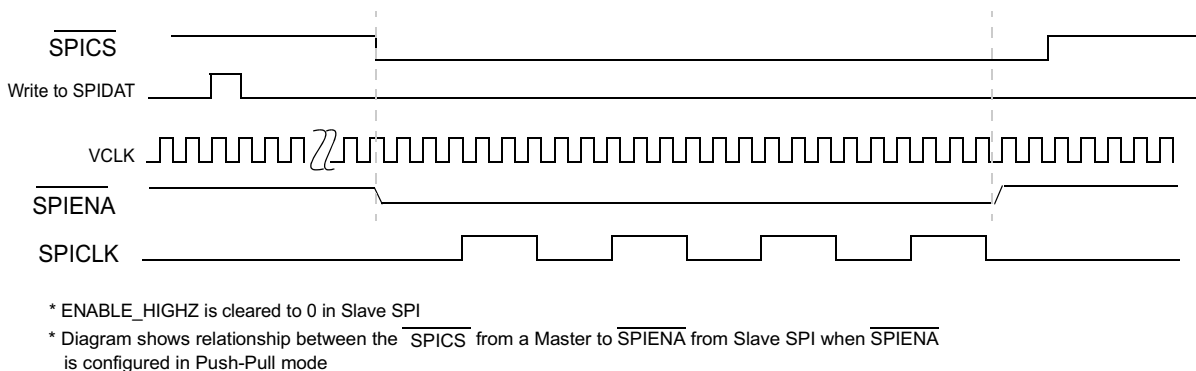
### 12.11.2 Peripheral Mode Timings for SPI/MibSPI



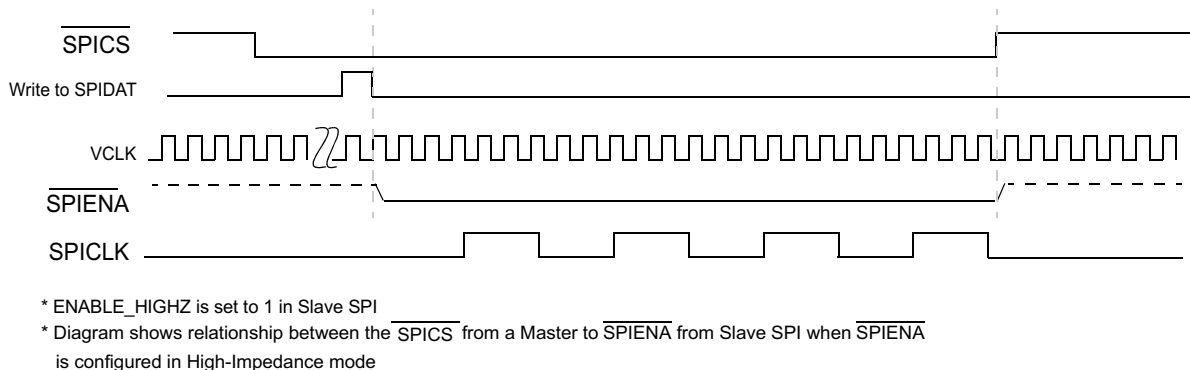
**Figure 12-85. SPI/MibSPI Pins During Peripheral Mode 3-pin Configuration**



**Figure 12-86. SPI/MibSPI Pins During Peripheral Mode 4-pin with  $\overline{\text{SPIENA}}$  Configuration**



**Figure 12-87. SPI/MibSPI Pins During Peripheral Mode in 5-pin Configuration - (Single Peripheral)**



**Figure 12-88. SPI/MibSPI Pins During Peripheral Mode in 5-pin Configuration - (Single/Multi Peripheral)**

### 12.11.3 Controller Timing Parameter Details

In case of Controller the module drives out SPICLK. It also drives out the Transmit data on SPISIMO with respect to its internal SPICLK. In case of Controller mode, the RX data on the SPISOMI pin is registered with respect to SPICLK received through the input buffer from the I/O pad.

If the chip select pin is functional, then the Controller will drive out the  $\overline{\text{SPICS}}$  pins before starting the SPICLK. If the  $\overline{\text{SPIENA}}$  pin is functional, then Controller will wait for an active low from the Peripheral on the input pin to start the SPICLK.

### 12.11.4 Peripheral Mode Timing Parameter Details

In case of Peripheral mode, the module will drive only the SPISOMI and  $\overline{\text{SPIENA}}$  pins. All other pins are inputs to it. The RX data on the SPISIMO pin will be registered with respect to the SPICLK pin. The Peripheral will use the  $\overline{\text{SPICS}}$  pin to drive out the  $\overline{\text{SPIENA}}$  pin if both are functional. If 4-pin with  $\overline{\text{SPIENA}}$  is configured, then the Peripheral will drive out an active-low signal on the  $\overline{\text{SPIENA}}$  pin when new data is written to the TX Shift Register. Irrespective of 4-pin with  $\overline{\text{SPIENA}}$  or 5-pin configuration, the Peripheral will deassert the  $\overline{\text{SPIENA}}$  pin after the last bit is received. If ENABLE\_HIGHZ (SPIINT0.24) bit is 0, the deasserted value of the  $\overline{\text{SPIENA}}$  pin will be 1. Otherwise, it will depend upon the internal pull up or pull down resistor (if implemented) depending upon the Specification of the Chip.

## Chapter 13

# ***Vectored Interrupt Manager (VIM) Module***

---



This chapter describes the behavior of the vectored interrupt manager (VIM) module of the device family.

<b>13.1 VIM Overview</b> .....	<b>3834</b>
<b>13.2 VIM Interrupt Inputs</b> .....	<b>3834</b>
<b>13.3 VIM Interrupt Outputs</b> .....	<b>3834</b>
<b>13.4 VIM Interrupt Vector Table (VIM RAM)</b> .....	<b>3834</b>
<b>13.5 VIM Interrupt Prioritization</b> .....	<b>3835</b>
<b>13.6 VIM ECC Support</b> .....	<b>3835</b>
<b>13.7 VIM Lockstep Mode</b> .....	<b>3836</b>
<b>13.8 VIM IDLE State</b> .....	<b>3836</b>
<b>13.9 VIM Interrupt Handling</b> .....	<b>3836</b>
<b>13.10 MSS_VIM Registers</b> .....	<b>3840</b>

## 13.1 VIM Overview

The VIM aggregates device interrupts and sends them to the R5F CPU(s). It can be used in either split or lockstep configuration. In split, it has two independent interrupt cores, one per CPU. In lockstep, CPU1 acts as a diagnostic on CPU0; only CPU0's outputs are used but all outputs are compared to CPU1 to provide diagnostic coverage.

The VIM module supports the following features:

- 256 interrupt inputs per R5F core
- Each interrupt has its own 4-bit programmable priority
  - Defined via the INTPRIORITY\_j register
  - The VIM provides support for priority interruption of interrupts
- Each interrupt has its own enable mask
  - Interrupt enable is done via the INTR\_EN\_SET\_j register
  - Interrupt disable is done via the INTR\_EN\_CLR\_j register
- Each interrupt can be programmed as either an IRQ or FIQ
  - Defined via the INTMAP\_j register
- Each interrupt has its own programmable 32-bit vector address associated with it
  - Defined via the INTVECTOR\_j register
  - Protected with SECED
- One IRQn and one FIQn output per core
- Vectored interrupt interface
  - Compatible with R5F VIC port
- Default vector provided when a double-bit error is detected
- Split or lockstep capable
  - In lockstep mode, only interrupts connected to VIM interrupt core 0 are available
- Software interrupt generation

## 13.2 VIM Interrupt Inputs

The VIM supports 256 interrupt inputs per core. Each interrupt can be either a level or a pulse (both active-high). The interrupt mapping for the two R5F cores can be found in [Chapter 7, Interrupts](#).

## 13.3 VIM Interrupt Outputs

The VIM has two interrupt outputs per core:

- *CoreN\_IRQn*: This is a normal interrupt for core *N* (active-low level). It can be serviced via the VIC interface or through the MMR interface. Whenever an interrupt input goes high, if that interrupt is mapped as an IRQ (via the INTMAP\_j register) and is enabled (via the INTR\_EN\_SET\_j register), then it will cause an IRQ to assert
- *CoreN\_FIQn*: This is a fast (or non-maskable) interrupt for core *N* (active-low level). FIQs always have priority over IRQs. An FIQ can be serviced through the MMR interface. Whenever an interrupt input goes high, if that interrupt is mapped as an FIQ and is enabled, then it will cause an FIQ to assert

## 13.4 VIM Interrupt Vector Table (VIM RAM)

For each VIM interrupt core, there is an associated interrupt vector table (VIM RAM) that is used to store the address of ISRs. During register vectored interrupt and hardware vectored interrupt, VIM accesses the interrupt vector table using the vector value to fetch the address of the corresponding ISR. Note that both interrupt vector tables are identical in their memory organization.

The VIM RAM is basically comprised of a set of interrupt vector registers (INTVECTOR\_j). Hence, the interrupt vector table is organized in 256 words of 30 bits, with a base address corresponding to the physical address of the first register in the group.



### Note

The lower two bits of the 32-bit interrupt vector are always 0s.

Figure 13-1 shows the VIM RAM interrupt vector map.

VIM RAM Address Space	VIM RAM Entries
Base Address + 0h	Interrupt 0 Vector
Base Address + 4h	Interrupt 1 Vector
Base Address + 8h	Interrupt 2 Vector
⋮	⋮
Base Address + 3F8h	Interrupt 254 Vector
Base Address + 3FCh	Interrupt 255 Vector

**Figure 13-1. VIM RAM Interrupt Vector Map**

The interrupt vector table has protection by ECC to indicate corruption due to soft errors. The ECC logic inside VIM supports SECDED. Refer to the ECC aggregator map for the VIM RAM ID.

## 13.5 VIM Interrupt Prioritization

The VIM supports the interruption of the currently active interrupt by one with a higher priority. FIQs and IRQs are completely separate but both use the same mechanism.

When an interrupt goes from pending to active (FIQ: reading the FIQVEC register; IRQ: reading the IRQVEC register, then the interrupt is loaded into the corresponding active register (ACTFIQ / ACTIRQ), and all interrupts of an equal or lesser priority are masked (discarded). If prior to this interrupt being cleared (by writing to the FIQVEC register, or IRQVEC register) another interrupt of higher priority arrives, then the FIQn/IRQn will be asserted and that interrupt made pending as normal. If the CPU switches this interrupt to active (by reading the FIQVEC / IRQVEC register), then the currently active interrupt will be pushed onto a stack. When an interrupt is cleared by reading the FIQVEC / IRQVEC register, if there are any interrupts on the stack, the first entry is popped off and put back into the ACTFIQ / ACTIRQ register, so that software may continue where it left off.

## 13.6 VIM ECC Support

The memory that holds the interrupt vector for each interrupt is protected by SECDED ECC. Single-bit errors are corrected and written back. Double-bit errors are not corrected. If a double-bit error occurs while trying to load a vector, then the DEDVEC register is used to provide the default vector for the *coreN\_IRQADDRV* signal, the IRQVEC register, and the FIQVEC register. The DEDVEC should point to an ISR that handles the fact that there was an uncorrectable error in the interrupt handling.

Some possible remediating actions would be to:

1. Reconstruct the vector table and re-start the application
  - a. Potentially switch to a completely software interrupt handler in the mean time
2. Restart the application from scratch
3. Reset the device

4. Sit in a loop (or WFI) while something external (for example, the ESM) responds to the DED interrupt that will be generated

It is up to the user and the application to determine the appropriate action.

---

#### Note

An interrupt that has an uncorrectable vector error (and thus uses the DED vector) will still have the priority of the original interrupt. This makes it possible for a higher priority interrupt to supercede the handling of the error.

Control and reporting are done by the ECC aggregator.

---

### 13.7 VIM Lockstep Mode

In lockstep mode, CPU1 is used as a diagnostic for CPU0. In this mode, only the interrupt inputs for CPU0 are used. Besides to CPU0, these interrupt inputs are also internally routed to CPU1 (through the level-sync / edge-detect logic dedicated to CPU1, and additionally through some delay circuits). The outputs from both VIM interrupt cores are then sent to the MSS CCMR5 module through dedicated compare buses (with CPU0's outputs delayed). The CCMR5 module is responsible for comparing the two sets of output signals and for reporting any mismatches by generating an interrupt (MSS\_CCMR5\_ERR).

---

#### Note

In lockstep mode, only the RAM dedicated to CPU0 is used, so software *must not* do anything with the ECC interface on the RAM dedicated to CPU1.

---

### 13.8 VIM IDLE State

The VIM will indicate IDLE when there are no pending unmasked interrupts or MMR accesses. The VIM does not have a clock stop interface.

### 13.9 VIM Interrupt Handling

There are multiple ways to service an interrupt depending on how much of the hardware assistance offered by the VIM the software wants to take advantage of.

For IRQs, it is recommended to use the procedure in [Section 13.9.1](#), but the procedures in [Section 13.9.2](#) or [Section 13.9.3](#) (if a user wants to implement a fully software prioritization scheme) may be used as alternatives.

For FIQs, it is recommended to use the procedure in [Section 13.9.4](#), but the procedure in [Section 13.9.5](#) may be used as an alternative.

---

#### Note

These descriptions do not include steps such as stack pushes and state retention that software must take in order to return from the ISR. It is assumed that the programmer is aware of these steps.

---

#### 13.9.1 Servicing IRQ Through Vector Interface

If the associated CPU has the vector (VIC) interface enabled, then the following method is used for servicing IRQs:

1. Hardware handshake
  - a. CPU asserts *coreN\_IRQACK* high
  - b. VIM asserts *coreN\_IRQADDRV* to indicate that the *coreN\_IRQADDR* bus is stable with the correct vector address
  - c. CPU reads *coreN\_IRQADDR*, jumps to that address, and de-asserts *coreN\_IRQACK* low
  - d. VIM de-asserts *coreN\_IRQn* and *coreN\_IRQADDRV*, VIM masks (discards) all IRQs with the same or lower priority

- e. VIM loads the value from the PRIIRQ[9:0] NUM bit field (which corresponds to the vector address) into the ACTIRQ[9:0] NUM bit field, which causes the ACTIRQ[31] VALID bit to be set
2. Service the interrupt
3. Depending on whether the original source of the interrupt was a pulse or a level (determined by reading the ACTIRQ[9:0] NUM bit field to determine number, and reading the appropriate bit in the INTTYPE\_j register to determine type)
  - a. Pulse
    - i. Clear the status by writing a '1' to the appropriate bit in the IRQSTS\_j register, or STS\_j register
    - ii. Clear the interrupt at the source. This way, the source can generate another pulse, if it needs to, and the VIM will process this as a new interrupt
  - b. Level
    - i. Clear the interrupt at the source
    - ii. Clear the status by writing a '1' to the appropriate bit in the IRQSTS\_j register, or STS\_j register. This way, the level should be gone at the input to the VIM, it will avoid falsely re-calling the interrupt. If the source maintains the level, then it means there is another interrupt
4. Write any value to the IRQVEC register
  - a. This will clear the priority mask and will cause all interrupts to be re-evaluated for the new highest priority interrupt
  - b. This will also clear the ACTIRQ[31] VALID bit

### 13.9.2 Servicing IRQ Through MMR Interface

When an IRQ interrupt is received, the CPU should follow these steps if not using the vector interface:

1. Read the IRQVEC register and jump to that address to service the ISR
  - a. Reading this register will mask (discard) all interrupts of an equal or lower priority and de-assert the *coreN\_IRQn* output. If another interrupt of a higher priority becomes available, the *coreN\_IRQn* will re-assert, allowing priority interruption of an interrupt
  - b. Reading this register will cause the value from the PRIIRQ[9:0] NUM bit field to be loaded into the ACTIRQ[9:0] NUM bit field, and the ACTIRQ[31] VALID bit to be set
2. Service the interrupt
3. Depending on whether the original source of the interrupt was a pulse or a level
  - a. Pulse
    - i. Clear the status by writing a '1' to the appropriate bit in the STS\_j register, or IRQSTS\_j register
    - ii. Clear the interrupt at the source
  - b. Level
    - i. Clear the interrupt at the source
    - ii. Clear the status by writing a '1' to the appropriate bit in the STS\_j register, or IRQSTS\_j register
4. Write any value to the IRQVEC register
  - a. This will clear the priority mask and will cause all interrupts to be re-evaluated for the new highest priority interrupt
  - b. This will also clear the ACTIRQ[31] VALID bit

### 13.9.3 Servicing IRQ Through MMR Interface (Alternative)

If a user does not want to use the IRQVEC register, the VIM may be used as a more traditional interrupt controller. Note that in this mode, there is no hardware priority masking (because the IRQVEC register is never read). Software would be responsible for doing all priority operations.

1. Determine which interrupt to service

- a. Read the PRIIRQ register to determine which interrupt is the highest priority IRQ currently asserted, OR
  - b. Optionally read the IRQGSTS register to determine which groups have IRQs pending, then read the IRQSTS\_j register and use a software prioritization scheme to determine which IRQ to service
2. Service the interrupt
3. Depending on whether the original source of the interrupt was a pulse or a level
  - a. Pulse
    - i. Clear the status by writing a '1' to the appropriate bit in the STS\_j register, or IRQSTS\_j register
    - ii. Clear the interrupt at the source.
  - b. Level
    - i. Clear the interrupt at the source
    - ii. Clear the status by writing a '1' to the appropriate bit in the STS\_j register, or IRQSTS\_j register

### 13.9.4 Servicing FIQ

When an FIQ interrupt is received, the CPU should follow these steps:

1. Read the FIQVEC register and jump to that address to service the ISR
  - a. Reading this register will mask (discard) all interrupts of an equal or lower priority and de-assert the *coreN\_FIQn* output. If another interrupt of a higher priority becomes available, the *coreN\_FIQn* will re-assert, allowing priority interruption of an interrupt.
  - b. Reading this register will cause the value from the PRIFIQ[9:0] NUM bit field to be loaded into the ACTFIQ[9:0] NUM bit field, and the ACTFIQ[31] VALID bit to be set
2. Service the interrupt
3. Depending on whether the original source of the interrupt was a pulse or a level (determined by reading the ACTFIQ[9:0] NUM bit field to determine number, and reading the appropriate bit in the INTTYPE\_j register to determine type)
  - a. Pulse
    - i. Clear the status by writing a '1' to the appropriate bit in the STS\_j register, or FIQSTS\_j register
    - ii. Clear the interrupt at the source. This way, the source can generate another pulse, if it needs to, and the VIM will process this as a new interrupt
  - b. Level
    - i. Clear the interrupt at the source
    - ii. Clear the status by writing a '1' to the appropriate bit in the STS\_j register, or FIQSTS\_j register. This way, the level should be gone at the input to the VIM, it will avoid falsely re-calling the interrupt. If the source maintains the level, then it means there is another interrupt
4. Write any value to the FIQVEC register
  - a. This will clear the priority mask and will cause all interrupts to be re-evaluated for the new highest priority interrupt
  - b. This will also clear the ACTFIQ[31] VALID bit

### 13.9.5 Servicing FIQ (Alternative)

If a user does not want to use the FIQVEC register, the VIM may be used as a more traditional interrupt controller. Note that in this mode, there is no hardware priority masking (because the FIQVEC register is never read). Software would be responsible for doing all priority operations.

1. Determine which interrupt to service
  - a. Read the PRIFIQ register to determine which interrupt is the highest priority FIQ currently asserted, OR
  - b. Optionally read the FIQGSTS register to determine which groups have IRQs pending, then read the FIQSTS\_j register and use a software prioritization scheme to determine which FIQ to service

2. Service the interrupt
3. Depending on whether the original source of the interrupt was a pulse or a level
  - a. Pulse
    - i. Clear the status by writing a '1' to the appropriate bit in the STS<sub>j</sub> register, or FIQSTS<sub>j</sub> register
    - ii. Clear the interrupt at the source.
  - b. Level
    - i. Clear the interrupt at the source
    - ii. Clear the status by writing a '1' to the appropriate bit in the STS<sub>j</sub> register, or FIQSTS<sub>j</sub> register.

## 13.10 MSS\_VIM Registers

Table 13-1 lists the memory-mapped registers for the MSS\_VIM registers. All register offset addresses not listed in Table 13-1 should be considered as reserved locations and the register contents should not be modified.

**Table 13-1. MSS\_VIM Registers**

Offset	Acronym	Register Name	Section
0h	PID	PID	<a href="#">Go</a>
4h	INFO	INFO	<a href="#">Go</a>
8h	PRIIRQ	PRIIRQ	<a href="#">Go</a>
Ch	PRIFIQ	PRIFIQ	<a href="#">Go</a>
10h	IRQGSTS	IRQGSTS	<a href="#">Go</a>
14h	FIQGSTS	FIQGSTS	<a href="#">Go</a>
18h	IRQVEC	IRQVEC	<a href="#">Go</a>
1Ch	FIQVEC	FIQVEC	<a href="#">Go</a>
20h	ACTIRQ	ACTIRQ	<a href="#">Go</a>
24h	ACTFIQ	ACTFIQ	<a href="#">Go</a>
30h	DEDVEC	DEDVEC	<a href="#">Go</a>
400h	RAW	RAW	<a href="#">Go</a>
404h	STS	STS	<a href="#">Go</a>
408h	INTR_EN_SET	INTR_EN_SET	<a href="#">Go</a>
40Ch	INTER_EN_CLR	INTER_EN_CLR	<a href="#">Go</a>
410h	IRQSTS	IRQSTS	<a href="#">Go</a>
414h	FIQSTS	FIQSTS	<a href="#">Go</a>
418h	INTMAP	INTMAP	<a href="#">Go</a>
41Ch	INTTYPE	INTTYPE	<a href="#">Go</a>
420h	RAW_1	RAW	<a href="#">Go</a>
424h	STS_1	STS	<a href="#">Go</a>
428h	INTR_EN_SET_1	INTR_EN_SET	<a href="#">Go</a>
42Ch	INTER_EN_CLR_1	INTER_EN_CLR	<a href="#">Go</a>
430h	IRQSTS_1	IRQSTS	<a href="#">Go</a>
434h	FIQSTS_1	FIQSTS	<a href="#">Go</a>
438h	INTMAP_1	INTMAP	<a href="#">Go</a>
43Ch	INTTYPE_1	INTTYPE	<a href="#">Go</a>
440h	RAW_2	RAW	<a href="#">Go</a>
444h	STS_2	STS	<a href="#">Go</a>
448h	INTR_EN_SET_2	INTR_EN_SET	<a href="#">Go</a>
44Ch	INTER_EN_CLR_2	INTER_EN_CLR	<a href="#">Go</a>
450h	IRQSTS_2	IRQSTS	<a href="#">Go</a>
454h	FIQSTS_2	FIQSTS	<a href="#">Go</a>
458h	INTMAP_2	INTMAP	<a href="#">Go</a>
45Ch	INTTYPE_2	INTTYPE	<a href="#">Go</a>
460h	RAW_3	RAW	<a href="#">Go</a>
464h	STS_3	STS	<a href="#">Go</a>
468h	INTR_EN_SET_3	INTR_EN_SET	<a href="#">Go</a>
46Ch	INTER_EN_CLR_3	INTER_EN_CLR	<a href="#">Go</a>
470h	IRQSTS_3	IRQSTS	<a href="#">Go</a>
474h	FIQSTS_3	FIQSTS	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
478h	INTMAP_3	INTMAP	<a href="#">Go</a>
47Ch	INTTYPE_3	INTTYPE	<a href="#">Go</a>
480h	RAW_4	RAW	<a href="#">Go</a>
484h	STS_4	STS	<a href="#">Go</a>
488h	INTR_EN_SET_4	INTR_EN_SET	<a href="#">Go</a>
48Ch	INTER_EN_CLR_4	INTER_EN_CLR	<a href="#">Go</a>
490h	IRQSTS_4	IRQSTS	<a href="#">Go</a>
494h	FIQSTS_4	FIQSTS	<a href="#">Go</a>
498h	INTMAP_4	INTMAP	<a href="#">Go</a>
49Ch	INTTYPE_4	INTTYPE	<a href="#">Go</a>
4A0h	RAW_5	RAW	<a href="#">Go</a>
4A4h	STS_5	STS	<a href="#">Go</a>
4A8h	INTR_EN_SET_5	INTR_EN_SET	<a href="#">Go</a>
4ACh	INTER_EN_CLR_5	INTER_EN_CLR	<a href="#">Go</a>
4B0h	IRQSTS_5	IRQSTS	<a href="#">Go</a>
4B4h	FIQSTS_5	FIQSTS	<a href="#">Go</a>
4B8h	INTMAP_5	INTMAP	<a href="#">Go</a>
4BCh	INTTYPE_5	INTTYPE	<a href="#">Go</a>
4C0h	RAW_6	RAW	<a href="#">Go</a>
4C4h	STS_6	STS	<a href="#">Go</a>
4C8h	INTR_EN_SET_6	INTR_EN_SET	<a href="#">Go</a>
4CCh	INTER_EN_CLR_6	INTER_EN_CLR	<a href="#">Go</a>
4D0h	IRQSTS_6	IRQSTS	<a href="#">Go</a>
4D4h	FIQSTS_6	FIQSTS	<a href="#">Go</a>
4D8h	INTMAP_6	INTMAP	<a href="#">Go</a>
4DCh	INTTYPE_6	INTTYPE	<a href="#">Go</a>
4E0h	RAW_7	RAW	<a href="#">Go</a>
4E4h	STS_7	STS	<a href="#">Go</a>
4E8h	INTR_EN_SET_7	INTR_EN_SET	<a href="#">Go</a>
4ECh	INTER_EN_CLR_7	INTER_EN_CLR	<a href="#">Go</a>
4F0h	IRQSTS_7	IRQSTS	<a href="#">Go</a>
4F4h	FIQSTS_7	FIQSTS	<a href="#">Go</a>
4F8h	INTMAP_7	INTMAP	<a href="#">Go</a>
4FCh	INTTYPE_7	INTTYPE	<a href="#">Go</a>
1000h	INTPRIORITY	INTPRIORITY	<a href="#">Go</a>
1004h	INTPRIORITY_1	INTPRIORITY	<a href="#">Go</a>
1008h	INTPRIORITY_2	INTPRIORITY	<a href="#">Go</a>
100Ch	INTPRIORITY_3	INTPRIORITY	<a href="#">Go</a>
1010h	INTPRIORITY_4	INTPRIORITY	<a href="#">Go</a>
1014h	INTPRIORITY_5	INTPRIORITY	<a href="#">Go</a>
1018h	INTPRIORITY_6	INTPRIORITY	<a href="#">Go</a>
101Ch	INTPRIORITY_7	INTPRIORITY	<a href="#">Go</a>
1020h	INTPRIORITY_8	INTPRIORITY	<a href="#">Go</a>
1024h	INTPRIORITY_9	INTPRIORITY	<a href="#">Go</a>
1028h	INTPRIORITY_10	INTPRIORITY	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
102Ch	INTPRIORITY_11	INTPRIORITY	<a href="#">Go</a>
1030h	INTPRIORITY_12	INTPRIORITY	<a href="#">Go</a>
1034h	INTPRIORITY_13	INTPRIORITY	<a href="#">Go</a>
1038h	INTPRIORITY_14	INTPRIORITY	<a href="#">Go</a>
103Ch	INTPRIORITY_15	INTPRIORITY	<a href="#">Go</a>
1040h	INTPRIORITY_16	INTPRIORITY	<a href="#">Go</a>
1044h	INTPRIORITY_17	INTPRIORITY	<a href="#">Go</a>
1048h	INTPRIORITY_18	INTPRIORITY	<a href="#">Go</a>
104Ch	INTPRIORITY_19	INTPRIORITY	<a href="#">Go</a>
1050h	INTPRIORITY_20	INTPRIORITY	<a href="#">Go</a>
1054h	INTPRIORITY_21	INTPRIORITY	<a href="#">Go</a>
1058h	INTPRIORITY_22	INTPRIORITY	<a href="#">Go</a>
105Ch	INTPRIORITY_23	INTPRIORITY	<a href="#">Go</a>
1060h	INTPRIORITY_24	INTPRIORITY	<a href="#">Go</a>
1064h	INTPRIORITY_25	INTPRIORITY	<a href="#">Go</a>
1068h	INTPRIORITY_26	INTPRIORITY	<a href="#">Go</a>
106Ch	INTPRIORITY_27	INTPRIORITY	<a href="#">Go</a>
1070h	INTPRIORITY_28	INTPRIORITY	<a href="#">Go</a>
1074h	INTPRIORITY_29	INTPRIORITY	<a href="#">Go</a>
1078h	INTPRIORITY_30	INTPRIORITY	<a href="#">Go</a>
107Ch	INTPRIORITY_31	INTPRIORITY	<a href="#">Go</a>
1080h	INTPRIORITY_32	INTPRIORITY	<a href="#">Go</a>
1084h	INTPRIORITY_33	INTPRIORITY	<a href="#">Go</a>
1088h	INTPRIORITY_34	INTPRIORITY	<a href="#">Go</a>
108Ch	INTPRIORITY_35	INTPRIORITY	<a href="#">Go</a>
1090h	INTPRIORITY_36	INTPRIORITY	<a href="#">Go</a>
1094h	INTPRIORITY_37	INTPRIORITY	<a href="#">Go</a>
1098h	INTPRIORITY_38	INTPRIORITY	<a href="#">Go</a>
109Ch	INTPRIORITY_39	INTPRIORITY	<a href="#">Go</a>
10A0h	INTPRIORITY_40	INTPRIORITY	<a href="#">Go</a>
10A4h	INTPRIORITY_41	INTPRIORITY	<a href="#">Go</a>
10A8h	INTPRIORITY_42	INTPRIORITY	<a href="#">Go</a>
10ACh	INTPRIORITY_43	INTPRIORITY	<a href="#">Go</a>
10B0h	INTPRIORITY_44	INTPRIORITY	<a href="#">Go</a>
10B4h	INTPRIORITY_45	INTPRIORITY	<a href="#">Go</a>
10B8h	INTPRIORITY_46	INTPRIORITY	<a href="#">Go</a>
10BCh	INTPRIORITY_47	INTPRIORITY	<a href="#">Go</a>
10C0h	INTPRIORITY_48	INTPRIORITY	<a href="#">Go</a>
10C4h	INTPRIORITY_49	INTPRIORITY	<a href="#">Go</a>
10C8h	INTPRIORITY_50	INTPRIORITY	<a href="#">Go</a>
10CCh	INTPRIORITY_51	INTPRIORITY	<a href="#">Go</a>
10D0h	INTPRIORITY_52	INTPRIORITY	<a href="#">Go</a>
10D4h	INTPRIORITY_53	INTPRIORITY	<a href="#">Go</a>
10D8h	INTPRIORITY_54	INTPRIORITY	<a href="#">Go</a>
10DCh	INTPRIORITY_55	INTPRIORITY	<a href="#">Go</a>



**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
10E0h	INTPRIORITY_56	INTPRIORITY	<a href="#">Go</a>
10E4h	INTPRIORITY_57	INTPRIORITY	<a href="#">Go</a>
10E8h	INTPRIORITY_58	INTPRIORITY	<a href="#">Go</a>
10ECh	INTPRIORITY_59	INTPRIORITY	<a href="#">Go</a>
10F0h	INTPRIORITY_60	INTPRIORITY	<a href="#">Go</a>
10F4h	INTPRIORITY_61	INTPRIORITY	<a href="#">Go</a>
10F8h	INTPRIORITY_62	INTPRIORITY	<a href="#">Go</a>
10FCh	INTPRIORITY_63	INTPRIORITY	<a href="#">Go</a>
1100h	INTPRIORITY_64	INTPRIORITY	<a href="#">Go</a>
1104h	INTPRIORITY_65	INTPRIORITY	<a href="#">Go</a>
1108h	INTPRIORITY_66	INTPRIORITY	<a href="#">Go</a>
110Ch	INTPRIORITY_67	INTPRIORITY	<a href="#">Go</a>
1110h	INTPRIORITY_68	INTPRIORITY	<a href="#">Go</a>
1114h	INTPRIORITY_69	INTPRIORITY	<a href="#">Go</a>
1118h	INTPRIORITY_70	INTPRIORITY	<a href="#">Go</a>
111Ch	INTPRIORITY_71	INTPRIORITY	<a href="#">Go</a>
1120h	INTPRIORITY_72	INTPRIORITY	<a href="#">Go</a>
1124h	INTPRIORITY_73	INTPRIORITY	<a href="#">Go</a>
1128h	INTPRIORITY_74	INTPRIORITY	<a href="#">Go</a>
112Ch	INTPRIORITY_75	INTPRIORITY	<a href="#">Go</a>
1130h	INTPRIORITY_76	INTPRIORITY	<a href="#">Go</a>
1134h	INTPRIORITY_77	INTPRIORITY	<a href="#">Go</a>
1138h	INTPRIORITY_78	INTPRIORITY	<a href="#">Go</a>
113Ch	INTPRIORITY_79	INTPRIORITY	<a href="#">Go</a>
1140h	INTPRIORITY_80	INTPRIORITY	<a href="#">Go</a>
1144h	INTPRIORITY_81	INTPRIORITY	<a href="#">Go</a>
1148h	INTPRIORITY_82	INTPRIORITY	<a href="#">Go</a>
114Ch	INTPRIORITY_83	INTPRIORITY	<a href="#">Go</a>
1150h	INTPRIORITY_84	INTPRIORITY	<a href="#">Go</a>
1154h	INTPRIORITY_85	INTPRIORITY	<a href="#">Go</a>
1158h	INTPRIORITY_86	INTPRIORITY	<a href="#">Go</a>
115Ch	INTPRIORITY_87	INTPRIORITY	<a href="#">Go</a>
1160h	INTPRIORITY_88	INTPRIORITY	<a href="#">Go</a>
1164h	INTPRIORITY_89	INTPRIORITY	<a href="#">Go</a>
1168h	INTPRIORITY_90	INTPRIORITY	<a href="#">Go</a>
116Ch	INTPRIORITY_91	INTPRIORITY	<a href="#">Go</a>
1170h	INTPRIORITY_92	INTPRIORITY	<a href="#">Go</a>
1174h	INTPRIORITY_93	INTPRIORITY	<a href="#">Go</a>
1178h	INTPRIORITY_94	INTPRIORITY	<a href="#">Go</a>
117Ch	INTPRIORITY_95	INTPRIORITY	<a href="#">Go</a>
1180h	INTPRIORITY_96	INTPRIORITY	<a href="#">Go</a>
1184h	INTPRIORITY_97	INTPRIORITY	<a href="#">Go</a>
1188h	INTPRIORITY_98	INTPRIORITY	<a href="#">Go</a>
118Ch	INTPRIORITY_99	INTPRIORITY	<a href="#">Go</a>
1190h	INTPRIORITY_100	INTPRIORITY	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
1194h	INTPRIORITY_101	INTPRIORITY	<a href="#">Go</a>
1198h	INTPRIORITY_102	INTPRIORITY	<a href="#">Go</a>
119Ch	INTPRIORITY_103	INTPRIORITY	<a href="#">Go</a>
11A0h	INTPRIORITY_104	INTPRIORITY	<a href="#">Go</a>
11A4h	INTPRIORITY_105	INTPRIORITY	<a href="#">Go</a>
11A8h	INTPRIORITY_106	INTPRIORITY	<a href="#">Go</a>
11ACh	INTPRIORITY_107	INTPRIORITY	<a href="#">Go</a>
11B0h	INTPRIORITY_108	INTPRIORITY	<a href="#">Go</a>
11B4h	INTPRIORITY_109	INTPRIORITY	<a href="#">Go</a>
11B8h	INTPRIORITY_110	INTPRIORITY	<a href="#">Go</a>
11BCh	INTPRIORITY_111	INTPRIORITY	<a href="#">Go</a>
11C0h	INTPRIORITY_112	INTPRIORITY	<a href="#">Go</a>
11C4h	INTPRIORITY_113	INTPRIORITY	<a href="#">Go</a>
11C8h	INTPRIORITY_114	INTPRIORITY	<a href="#">Go</a>
11CCh	INTPRIORITY_115	INTPRIORITY	<a href="#">Go</a>
11D0h	INTPRIORITY_116	INTPRIORITY	<a href="#">Go</a>
11D4h	INTPRIORITY_117	INTPRIORITY	<a href="#">Go</a>
11D8h	INTPRIORITY_118	INTPRIORITY	<a href="#">Go</a>
11DCh	INTPRIORITY_119	INTPRIORITY	<a href="#">Go</a>
11E0h	INTPRIORITY_120	INTPRIORITY	<a href="#">Go</a>
11E4h	INTPRIORITY_121	INTPRIORITY	<a href="#">Go</a>
11E8h	INTPRIORITY_122	INTPRIORITY	<a href="#">Go</a>
11ECh	INTPRIORITY_123	INTPRIORITY	<a href="#">Go</a>
11F0h	INTPRIORITY_124	INTPRIORITY	<a href="#">Go</a>
11F4h	INTPRIORITY_125	INTPRIORITY	<a href="#">Go</a>
11F8h	INTPRIORITY_126	INTPRIORITY	<a href="#">Go</a>
11FCh	INTPRIORITY_127	INTPRIORITY	<a href="#">Go</a>
1200h	INTPRIORITY_128	INTPRIORITY	<a href="#">Go</a>
1204h	INTPRIORITY_129	INTPRIORITY	<a href="#">Go</a>
1208h	INTPRIORITY_130	INTPRIORITY	<a href="#">Go</a>
120Ch	INTPRIORITY_131	INTPRIORITY	<a href="#">Go</a>
1210h	INTPRIORITY_132	INTPRIORITY	<a href="#">Go</a>
1214h	INTPRIORITY_133	INTPRIORITY	<a href="#">Go</a>
1218h	INTPRIORITY_134	INTPRIORITY	<a href="#">Go</a>
121Ch	INTPRIORITY_135	INTPRIORITY	<a href="#">Go</a>
1220h	INTPRIORITY_136	INTPRIORITY	<a href="#">Go</a>
1224h	INTPRIORITY_137	INTPRIORITY	<a href="#">Go</a>
1228h	INTPRIORITY_138	INTPRIORITY	<a href="#">Go</a>
122Ch	INTPRIORITY_139	INTPRIORITY	<a href="#">Go</a>
1230h	INTPRIORITY_140	INTPRIORITY	<a href="#">Go</a>
1234h	INTPRIORITY_141	INTPRIORITY	<a href="#">Go</a>
1238h	INTPRIORITY_142	INTPRIORITY	<a href="#">Go</a>
123Ch	INTPRIORITY_143	INTPRIORITY	<a href="#">Go</a>
1240h	INTPRIORITY_144	INTPRIORITY	<a href="#">Go</a>
1244h	INTPRIORITY_145	INTPRIORITY	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
1248h	INTPRIORITY_146	INTPRIORITY	<a href="#">Go</a>
124Ch	INTPRIORITY_147	INTPRIORITY	<a href="#">Go</a>
1250h	INTPRIORITY_148	INTPRIORITY	<a href="#">Go</a>
1254h	INTPRIORITY_149	INTPRIORITY	<a href="#">Go</a>
1258h	INTPRIORITY_150	INTPRIORITY	<a href="#">Go</a>
125Ch	INTPRIORITY_151	INTPRIORITY	<a href="#">Go</a>
1260h	INTPRIORITY_152	INTPRIORITY	<a href="#">Go</a>
1264h	INTPRIORITY_153	INTPRIORITY	<a href="#">Go</a>
1268h	INTPRIORITY_154	INTPRIORITY	<a href="#">Go</a>
126Ch	INTPRIORITY_155	INTPRIORITY	<a href="#">Go</a>
1270h	INTPRIORITY_156	INTPRIORITY	<a href="#">Go</a>
1274h	INTPRIORITY_157	INTPRIORITY	<a href="#">Go</a>
1278h	INTPRIORITY_158	INTPRIORITY	<a href="#">Go</a>
127Ch	INTPRIORITY_159	INTPRIORITY	<a href="#">Go</a>
1280h	INTPRIORITY_160	INTPRIORITY	<a href="#">Go</a>
1284h	INTPRIORITY_161	INTPRIORITY	<a href="#">Go</a>
1288h	INTPRIORITY_162	INTPRIORITY	<a href="#">Go</a>
128Ch	INTPRIORITY_163	INTPRIORITY	<a href="#">Go</a>
1290h	INTPRIORITY_164	INTPRIORITY	<a href="#">Go</a>
1294h	INTPRIORITY_165	INTPRIORITY	<a href="#">Go</a>
1298h	INTPRIORITY_166	INTPRIORITY	<a href="#">Go</a>
129Ch	INTPRIORITY_167	INTPRIORITY	<a href="#">Go</a>
12A0h	INTPRIORITY_168	INTPRIORITY	<a href="#">Go</a>
12A4h	INTPRIORITY_169	INTPRIORITY	<a href="#">Go</a>
12A8h	INTPRIORITY_170	INTPRIORITY	<a href="#">Go</a>
12ACh	INTPRIORITY_171	INTPRIORITY	<a href="#">Go</a>
12B0h	INTPRIORITY_172	INTPRIORITY	<a href="#">Go</a>
12B4h	INTPRIORITY_173	INTPRIORITY	<a href="#">Go</a>
12B8h	INTPRIORITY_174	INTPRIORITY	<a href="#">Go</a>
12BCh	INTPRIORITY_175	INTPRIORITY	<a href="#">Go</a>
12C0h	INTPRIORITY_176	INTPRIORITY	<a href="#">Go</a>
12C4h	INTPRIORITY_177	INTPRIORITY	<a href="#">Go</a>
12C8h	INTPRIORITY_178	INTPRIORITY	<a href="#">Go</a>
12CCh	INTPRIORITY_179	INTPRIORITY	<a href="#">Go</a>
12D0h	INTPRIORITY_180	INTPRIORITY	<a href="#">Go</a>
12D4h	INTPRIORITY_181	INTPRIORITY	<a href="#">Go</a>
12D8h	INTPRIORITY_182	INTPRIORITY	<a href="#">Go</a>
12DCh	INTPRIORITY_183	INTPRIORITY	<a href="#">Go</a>
12E0h	INTPRIORITY_184	INTPRIORITY	<a href="#">Go</a>
12E4h	INTPRIORITY_185	INTPRIORITY	<a href="#">Go</a>
12E8h	INTPRIORITY_186	INTPRIORITY	<a href="#">Go</a>
12ECh	INTPRIORITY_187	INTPRIORITY	<a href="#">Go</a>
12F0h	INTPRIORITY_188	INTPRIORITY	<a href="#">Go</a>
12F4h	INTPRIORITY_189	INTPRIORITY	<a href="#">Go</a>
12F8h	INTPRIORITY_190	INTPRIORITY	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
12FCh	INTPRIORITY_191	INTPRIORITY	<a href="#">Go</a>
1300h	INTPRIORITY_192	INTPRIORITY	<a href="#">Go</a>
1304h	INTPRIORITY_193	INTPRIORITY	<a href="#">Go</a>
1308h	INTPRIORITY_194	INTPRIORITY	<a href="#">Go</a>
130Ch	INTPRIORITY_195	INTPRIORITY	<a href="#">Go</a>
1310h	INTPRIORITY_196	INTPRIORITY	<a href="#">Go</a>
1314h	INTPRIORITY_197	INTPRIORITY	<a href="#">Go</a>
1318h	INTPRIORITY_198	INTPRIORITY	<a href="#">Go</a>
131Ch	INTPRIORITY_199	INTPRIORITY	<a href="#">Go</a>
1320h	INTPRIORITY_200	INTPRIORITY	<a href="#">Go</a>
1324h	INTPRIORITY_201	INTPRIORITY	<a href="#">Go</a>
1328h	INTPRIORITY_202	INTPRIORITY	<a href="#">Go</a>
132Ch	INTPRIORITY_203	INTPRIORITY	<a href="#">Go</a>
1330h	INTPRIORITY_204	INTPRIORITY	<a href="#">Go</a>
1334h	INTPRIORITY_205	INTPRIORITY	<a href="#">Go</a>
1338h	INTPRIORITY_206	INTPRIORITY	<a href="#">Go</a>
133Ch	INTPRIORITY_207	INTPRIORITY	<a href="#">Go</a>
1340h	INTPRIORITY_208	INTPRIORITY	<a href="#">Go</a>
1344h	INTPRIORITY_209	INTPRIORITY	<a href="#">Go</a>
1348h	INTPRIORITY_210	INTPRIORITY	<a href="#">Go</a>
134Ch	INTPRIORITY_211	INTPRIORITY	<a href="#">Go</a>
1350h	INTPRIORITY_212	INTPRIORITY	<a href="#">Go</a>
1354h	INTPRIORITY_213	INTPRIORITY	<a href="#">Go</a>
1358h	INTPRIORITY_214	INTPRIORITY	<a href="#">Go</a>
135Ch	INTPRIORITY_215	INTPRIORITY	<a href="#">Go</a>
1360h	INTPRIORITY_216	INTPRIORITY	<a href="#">Go</a>
1364h	INTPRIORITY_217	INTPRIORITY	<a href="#">Go</a>
1368h	INTPRIORITY_218	INTPRIORITY	<a href="#">Go</a>
136Ch	INTPRIORITY_219	INTPRIORITY	<a href="#">Go</a>
1370h	INTPRIORITY_220	INTPRIORITY	<a href="#">Go</a>
1374h	INTPRIORITY_221	INTPRIORITY	<a href="#">Go</a>
1378h	INTPRIORITY_222	INTPRIORITY	<a href="#">Go</a>
137Ch	INTPRIORITY_223	INTPRIORITY	<a href="#">Go</a>
1380h	INTPRIORITY_224	INTPRIORITY	<a href="#">Go</a>
1384h	INTPRIORITY_225	INTPRIORITY	<a href="#">Go</a>
1388h	INTPRIORITY_226	INTPRIORITY	<a href="#">Go</a>
138Ch	INTPRIORITY_227	INTPRIORITY	<a href="#">Go</a>
1390h	INTPRIORITY_228	INTPRIORITY	<a href="#">Go</a>
1394h	INTPRIORITY_229	INTPRIORITY	<a href="#">Go</a>
1398h	INTPRIORITY_230	INTPRIORITY	<a href="#">Go</a>
139Ch	INTPRIORITY_231	INTPRIORITY	<a href="#">Go</a>
13A0h	INTPRIORITY_232	INTPRIORITY	<a href="#">Go</a>
13A4h	INTPRIORITY_233	INTPRIORITY	<a href="#">Go</a>
13A8h	INTPRIORITY_234	INTPRIORITY	<a href="#">Go</a>
13ACh	INTPRIORITY_235	INTPRIORITY	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
13B0h	INTPRIORITY_236	INTPRIORITY	<a href="#">Go</a>
13B4h	INTPRIORITY_237	INTPRIORITY	<a href="#">Go</a>
13B8h	INTPRIORITY_238	INTPRIORITY	<a href="#">Go</a>
13BCh	INTPRIORITY_239	INTPRIORITY	<a href="#">Go</a>
13C0h	INTPRIORITY_240	INTPRIORITY	<a href="#">Go</a>
13C4h	INTPRIORITY_241	INTPRIORITY	<a href="#">Go</a>
13C8h	INTPRIORITY_242	INTPRIORITY	<a href="#">Go</a>
13CCh	INTPRIORITY_243	INTPRIORITY	<a href="#">Go</a>
13D0h	INTPRIORITY_244	INTPRIORITY	<a href="#">Go</a>
13D4h	INTPRIORITY_245	INTPRIORITY	<a href="#">Go</a>
13D8h	INTPRIORITY_246	INTPRIORITY	<a href="#">Go</a>
13DCh	INTPRIORITY_247	INTPRIORITY	<a href="#">Go</a>
13E0h	INTPRIORITY_248	INTPRIORITY	<a href="#">Go</a>
13E4h	INTPRIORITY_249	INTPRIORITY	<a href="#">Go</a>
13E8h	INTPRIORITY_250	INTPRIORITY	<a href="#">Go</a>
13ECh	INTPRIORITY_251	INTPRIORITY	<a href="#">Go</a>
13F0h	INTPRIORITY_252	INTPRIORITY	<a href="#">Go</a>
13F4h	INTPRIORITY_253	INTPRIORITY	<a href="#">Go</a>
13F8h	INTPRIORITY_254	INTPRIORITY	<a href="#">Go</a>
13FCh	INTPRIORITY_255	INTPRIORITY	<a href="#">Go</a>
2000h	INTVECTOR	INTVECTOR	<a href="#">Go</a>
2004h	INTVECTOR_1	INTVECTOR	<a href="#">Go</a>
2008h	INTVECTOR_2	INTVECTOR	<a href="#">Go</a>
200Ch	INTVECTOR_3	INTVECTOR	<a href="#">Go</a>
2010h	INTVECTOR_4	INTVECTOR	<a href="#">Go</a>
2014h	INTVECTOR_5	INTVECTOR	<a href="#">Go</a>
2018h	INTVECTOR_6	INTVECTOR	<a href="#">Go</a>
201Ch	INTVECTOR_7	INTVECTOR	<a href="#">Go</a>
2020h	INTVECTOR_8	INTVECTOR	<a href="#">Go</a>
2024h	INTVECTOR_9	INTVECTOR	<a href="#">Go</a>
2028h	INTVECTOR_10	INTVECTOR	<a href="#">Go</a>
202Ch	INTVECTOR_11	INTVECTOR	<a href="#">Go</a>
2030h	INTVECTOR_12	INTVECTOR	<a href="#">Go</a>
2034h	INTVECTOR_13	INTVECTOR	<a href="#">Go</a>
2038h	INTVECTOR_14	INTVECTOR	<a href="#">Go</a>
203Ch	INTVECTOR_15	INTVECTOR	<a href="#">Go</a>
2040h	INTVECTOR_16	INTVECTOR	<a href="#">Go</a>
2044h	INTVECTOR_17	INTVECTOR	<a href="#">Go</a>
2048h	INTVECTOR_18	INTVECTOR	<a href="#">Go</a>
204Ch	INTVECTOR_19	INTVECTOR	<a href="#">Go</a>
2050h	INTVECTOR_20	INTVECTOR	<a href="#">Go</a>
2054h	INTVECTOR_21	INTVECTOR	<a href="#">Go</a>
2058h	INTVECTOR_22	INTVECTOR	<a href="#">Go</a>
205Ch	INTVECTOR_23	INTVECTOR	<a href="#">Go</a>
2060h	INTVECTOR_24	INTVECTOR	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
2064h	INTVECTOR_25	INTVECTOR	<a href="#">Go</a>
2068h	INTVECTOR_26	INTVECTOR	<a href="#">Go</a>
206Ch	INTVECTOR_27	INTVECTOR	<a href="#">Go</a>
2070h	INTVECTOR_28	INTVECTOR	<a href="#">Go</a>
2074h	INTVECTOR_29	INTVECTOR	<a href="#">Go</a>
2078h	INTVECTOR_30	INTVECTOR	<a href="#">Go</a>
207Ch	INTVECTOR_31	INTVECTOR	<a href="#">Go</a>
2080h	INTVECTOR_32	INTVECTOR	<a href="#">Go</a>
2084h	INTVECTOR_33	INTVECTOR	<a href="#">Go</a>
2088h	INTVECTOR_34	INTVECTOR	<a href="#">Go</a>
208Ch	INTVECTOR_35	INTVECTOR	<a href="#">Go</a>
2090h	INTVECTOR_36	INTVECTOR	<a href="#">Go</a>
2094h	INTVECTOR_37	INTVECTOR	<a href="#">Go</a>
2098h	INTVECTOR_38	INTVECTOR	<a href="#">Go</a>
209Ch	INTVECTOR_39	INTVECTOR	<a href="#">Go</a>
20A0h	INTVECTOR_40	INTVECTOR	<a href="#">Go</a>
20A4h	INTVECTOR_41	INTVECTOR	<a href="#">Go</a>
20A8h	INTVECTOR_42	INTVECTOR	<a href="#">Go</a>
20ACh	INTVECTOR_43	INTVECTOR	<a href="#">Go</a>
20B0h	INTVECTOR_44	INTVECTOR	<a href="#">Go</a>
20B4h	INTVECTOR_45	INTVECTOR	<a href="#">Go</a>
20B8h	INTVECTOR_46	INTVECTOR	<a href="#">Go</a>
20BCh	INTVECTOR_47	INTVECTOR	<a href="#">Go</a>
20C0h	INTVECTOR_48	INTVECTOR	<a href="#">Go</a>
20C4h	INTVECTOR_49	INTVECTOR	<a href="#">Go</a>
20C8h	INTVECTOR_50	INTVECTOR	<a href="#">Go</a>
20CCh	INTVECTOR_51	INTVECTOR	<a href="#">Go</a>
20D0h	INTVECTOR_52	INTVECTOR	<a href="#">Go</a>
20D4h	INTVECTOR_53	INTVECTOR	<a href="#">Go</a>
20D8h	INTVECTOR_54	INTVECTOR	<a href="#">Go</a>
20DCh	INTVECTOR_55	INTVECTOR	<a href="#">Go</a>
20E0h	INTVECTOR_56	INTVECTOR	<a href="#">Go</a>
20E4h	INTVECTOR_57	INTVECTOR	<a href="#">Go</a>
20E8h	INTVECTOR_58	INTVECTOR	<a href="#">Go</a>
20ECh	INTVECTOR_59	INTVECTOR	<a href="#">Go</a>
20F0h	INTVECTOR_60	INTVECTOR	<a href="#">Go</a>
20F4h	INTVECTOR_61	INTVECTOR	<a href="#">Go</a>
20F8h	INTVECTOR_62	INTVECTOR	<a href="#">Go</a>
20FCh	INTVECTOR_63	INTVECTOR	<a href="#">Go</a>
2100h	INTVECTOR_64	INTVECTOR	<a href="#">Go</a>
2104h	INTVECTOR_65	INTVECTOR	<a href="#">Go</a>
2108h	INTVECTOR_66	INTVECTOR	<a href="#">Go</a>
210Ch	INTVECTOR_67	INTVECTOR	<a href="#">Go</a>
2110h	INTVECTOR_68	INTVECTOR	<a href="#">Go</a>
2114h	INTVECTOR_69	INTVECTOR	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
2118h	INTVECTOR_70	INTVECTOR	<a href="#">Go</a>
211Ch	INTVECTOR_71	INTVECTOR	<a href="#">Go</a>
2120h	INTVECTOR_72	INTVECTOR	<a href="#">Go</a>
2124h	INTVECTOR_73	INTVECTOR	<a href="#">Go</a>
2128h	INTVECTOR_74	INTVECTOR	<a href="#">Go</a>
212Ch	INTVECTOR_75	INTVECTOR	<a href="#">Go</a>
2130h	INTVECTOR_76	INTVECTOR	<a href="#">Go</a>
2134h	INTVECTOR_77	INTVECTOR	<a href="#">Go</a>
2138h	INTVECTOR_78	INTVECTOR	<a href="#">Go</a>
213Ch	INTVECTOR_79	INTVECTOR	<a href="#">Go</a>
2140h	INTVECTOR_80	INTVECTOR	<a href="#">Go</a>
2144h	INTVECTOR_81	INTVECTOR	<a href="#">Go</a>
2148h	INTVECTOR_82	INTVECTOR	<a href="#">Go</a>
214Ch	INTVECTOR_83	INTVECTOR	<a href="#">Go</a>
2150h	INTVECTOR_84	INTVECTOR	<a href="#">Go</a>
2154h	INTVECTOR_85	INTVECTOR	<a href="#">Go</a>
2158h	INTVECTOR_86	INTVECTOR	<a href="#">Go</a>
215Ch	INTVECTOR_87	INTVECTOR	<a href="#">Go</a>
2160h	INTVECTOR_88	INTVECTOR	<a href="#">Go</a>
2164h	INTVECTOR_89	INTVECTOR	<a href="#">Go</a>
2168h	INTVECTOR_90	INTVECTOR	<a href="#">Go</a>
216Ch	INTVECTOR_91	INTVECTOR	<a href="#">Go</a>
2170h	INTVECTOR_92	INTVECTOR	<a href="#">Go</a>
2174h	INTVECTOR_93	INTVECTOR	<a href="#">Go</a>
2178h	INTVECTOR_94	INTVECTOR	<a href="#">Go</a>
217Ch	INTVECTOR_95	INTVECTOR	<a href="#">Go</a>
2180h	INTVECTOR_96	INTVECTOR	<a href="#">Go</a>
2184h	INTVECTOR_97	INTVECTOR	<a href="#">Go</a>
2188h	INTVECTOR_98	INTVECTOR	<a href="#">Go</a>
218Ch	INTVECTOR_99	INTVECTOR	<a href="#">Go</a>
2190h	INTVECTOR_100	INTVECTOR	<a href="#">Go</a>
2194h	INTVECTOR_101	INTVECTOR	<a href="#">Go</a>
2198h	INTVECTOR_102	INTVECTOR	<a href="#">Go</a>
219Ch	INTVECTOR_103	INTVECTOR	<a href="#">Go</a>
21A0h	INTVECTOR_104	INTVECTOR	<a href="#">Go</a>
21A4h	INTVECTOR_105	INTVECTOR	<a href="#">Go</a>
21A8h	INTVECTOR_106	INTVECTOR	<a href="#">Go</a>
21ACh	INTVECTOR_107	INTVECTOR	<a href="#">Go</a>
21B0h	INTVECTOR_108	INTVECTOR	<a href="#">Go</a>
21B4h	INTVECTOR_109	INTVECTOR	<a href="#">Go</a>
21B8h	INTVECTOR_110	INTVECTOR	<a href="#">Go</a>
21BCh	INTVECTOR_111	INTVECTOR	<a href="#">Go</a>
21C0h	INTVECTOR_112	INTVECTOR	<a href="#">Go</a>
21C4h	INTVECTOR_113	INTVECTOR	<a href="#">Go</a>
21C8h	INTVECTOR_114	INTVECTOR	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
21CCh	INTVECTOR_115	INTVECTOR	<a href="#">Go</a>
21D0h	INTVECTOR_116	INTVECTOR	<a href="#">Go</a>
21D4h	INTVECTOR_117	INTVECTOR	<a href="#">Go</a>
21D8h	INTVECTOR_118	INTVECTOR	<a href="#">Go</a>
21DCh	INTVECTOR_119	INTVECTOR	<a href="#">Go</a>
21E0h	INTVECTOR_120	INTVECTOR	<a href="#">Go</a>
21E4h	INTVECTOR_121	INTVECTOR	<a href="#">Go</a>
21E8h	INTVECTOR_122	INTVECTOR	<a href="#">Go</a>
21ECh	INTVECTOR_123	INTVECTOR	<a href="#">Go</a>
21F0h	INTVECTOR_124	INTVECTOR	<a href="#">Go</a>
21F4h	INTVECTOR_125	INTVECTOR	<a href="#">Go</a>
21F8h	INTVECTOR_126	INTVECTOR	<a href="#">Go</a>
21FCh	INTVECTOR_127	INTVECTOR	<a href="#">Go</a>
2200h	INTVECTOR_128	INTVECTOR	<a href="#">Go</a>
2204h	INTVECTOR_129	INTVECTOR	<a href="#">Go</a>
2208h	INTVECTOR_130	INTVECTOR	<a href="#">Go</a>
220Ch	INTVECTOR_131	INTVECTOR	<a href="#">Go</a>
2210h	INTVECTOR_132	INTVECTOR	<a href="#">Go</a>
2214h	INTVECTOR_133	INTVECTOR	<a href="#">Go</a>
2218h	INTVECTOR_134	INTVECTOR	<a href="#">Go</a>
221Ch	INTVECTOR_135	INTVECTOR	<a href="#">Go</a>
2220h	INTVECTOR_136	INTVECTOR	<a href="#">Go</a>
2224h	INTVECTOR_137	INTVECTOR	<a href="#">Go</a>
2228h	INTVECTOR_138	INTVECTOR	<a href="#">Go</a>
222Ch	INTVECTOR_139	INTVECTOR	<a href="#">Go</a>
2230h	INTVECTOR_140	INTVECTOR	<a href="#">Go</a>
2234h	INTVECTOR_141	INTVECTOR	<a href="#">Go</a>
2238h	INTVECTOR_142	INTVECTOR	<a href="#">Go</a>
223Ch	INTVECTOR_143	INTVECTOR	<a href="#">Go</a>
2240h	INTVECTOR_144	INTVECTOR	<a href="#">Go</a>
2244h	INTVECTOR_145	INTVECTOR	<a href="#">Go</a>
2248h	INTVECTOR_146	INTVECTOR	<a href="#">Go</a>
224Ch	INTVECTOR_147	INTVECTOR	<a href="#">Go</a>
2250h	INTVECTOR_148	INTVECTOR	<a href="#">Go</a>
2254h	INTVECTOR_149	INTVECTOR	<a href="#">Go</a>
2258h	INTVECTOR_150	INTVECTOR	<a href="#">Go</a>
225Ch	INTVECTOR_151	INTVECTOR	<a href="#">Go</a>
2260h	INTVECTOR_152	INTVECTOR	<a href="#">Go</a>
2264h	INTVECTOR_153	INTVECTOR	<a href="#">Go</a>
2268h	INTVECTOR_154	INTVECTOR	<a href="#">Go</a>
226Ch	INTVECTOR_155	INTVECTOR	<a href="#">Go</a>
2270h	INTVECTOR_156	INTVECTOR	<a href="#">Go</a>
2274h	INTVECTOR_157	INTVECTOR	<a href="#">Go</a>
2278h	INTVECTOR_158	INTVECTOR	<a href="#">Go</a>
227Ch	INTVECTOR_159	INTVECTOR	<a href="#">Go</a>



**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
2280h	INTVECTOR_160	INTVECTOR	<a href="#">Go</a>
2284h	INTVECTOR_161	INTVECTOR	<a href="#">Go</a>
2288h	INTVECTOR_162	INTVECTOR	<a href="#">Go</a>
228Ch	INTVECTOR_163	INTVECTOR	<a href="#">Go</a>
2290h	INTVECTOR_164	INTVECTOR	<a href="#">Go</a>
2294h	INTVECTOR_165	INTVECTOR	<a href="#">Go</a>
2298h	INTVECTOR_166	INTVECTOR	<a href="#">Go</a>
229Ch	INTVECTOR_167	INTVECTOR	<a href="#">Go</a>
22A0h	INTVECTOR_168	INTVECTOR	<a href="#">Go</a>
22A4h	INTVECTOR_169	INTVECTOR	<a href="#">Go</a>
22A8h	INTVECTOR_170	INTVECTOR	<a href="#">Go</a>
22ACh	INTVECTOR_171	INTVECTOR	<a href="#">Go</a>
22B0h	INTVECTOR_172	INTVECTOR	<a href="#">Go</a>
22B4h	INTVECTOR_173	INTVECTOR	<a href="#">Go</a>
22B8h	INTVECTOR_174	INTVECTOR	<a href="#">Go</a>
22BCh	INTVECTOR_175	INTVECTOR	<a href="#">Go</a>
22C0h	INTVECTOR_176	INTVECTOR	<a href="#">Go</a>
22C4h	INTVECTOR_177	INTVECTOR	<a href="#">Go</a>
22C8h	INTVECTOR_178	INTVECTOR	<a href="#">Go</a>
22CCh	INTVECTOR_179	INTVECTOR	<a href="#">Go</a>
22D0h	INTVECTOR_180	INTVECTOR	<a href="#">Go</a>
22D4h	INTVECTOR_181	INTVECTOR	<a href="#">Go</a>
22D8h	INTVECTOR_182	INTVECTOR	<a href="#">Go</a>
22DCh	INTVECTOR_183	INTVECTOR	<a href="#">Go</a>
22E0h	INTVECTOR_184	INTVECTOR	<a href="#">Go</a>
22E4h	INTVECTOR_185	INTVECTOR	<a href="#">Go</a>
22E8h	INTVECTOR_186	INTVECTOR	<a href="#">Go</a>
22ECh	INTVECTOR_187	INTVECTOR	<a href="#">Go</a>
22F0h	INTVECTOR_188	INTVECTOR	<a href="#">Go</a>
22F4h	INTVECTOR_189	INTVECTOR	<a href="#">Go</a>
22F8h	INTVECTOR_190	INTVECTOR	<a href="#">Go</a>
22FCh	INTVECTOR_191	INTVECTOR	<a href="#">Go</a>
2300h	INTVECTOR_192	INTVECTOR	<a href="#">Go</a>
2304h	INTVECTOR_193	INTVECTOR	<a href="#">Go</a>
2308h	INTVECTOR_194	INTVECTOR	<a href="#">Go</a>
230Ch	INTVECTOR_195	INTVECTOR	<a href="#">Go</a>
2310h	INTVECTOR_196	INTVECTOR	<a href="#">Go</a>
2314h	INTVECTOR_197	INTVECTOR	<a href="#">Go</a>
2318h	INTVECTOR_198	INTVECTOR	<a href="#">Go</a>
231Ch	INTVECTOR_199	INTVECTOR	<a href="#">Go</a>
2320h	INTVECTOR_200	INTVECTOR	<a href="#">Go</a>
2324h	INTVECTOR_201	INTVECTOR	<a href="#">Go</a>
2328h	INTVECTOR_202	INTVECTOR	<a href="#">Go</a>
232Ch	INTVECTOR_203	INTVECTOR	<a href="#">Go</a>
2330h	INTVECTOR_204	INTVECTOR	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
2334h	INTVECTOR_205	INTVECTOR	<a href="#">Go</a>
2338h	INTVECTOR_206	INTVECTOR	<a href="#">Go</a>
233Ch	INTVECTOR_207	INTVECTOR	<a href="#">Go</a>
2340h	INTVECTOR_208	INTVECTOR	<a href="#">Go</a>
2344h	INTVECTOR_209	INTVECTOR	<a href="#">Go</a>
2348h	INTVECTOR_210	INTVECTOR	<a href="#">Go</a>
234Ch	INTVECTOR_211	INTVECTOR	<a href="#">Go</a>
2350h	INTVECTOR_212	INTVECTOR	<a href="#">Go</a>
2354h	INTVECTOR_213	INTVECTOR	<a href="#">Go</a>
2358h	INTVECTOR_214	INTVECTOR	<a href="#">Go</a>
235Ch	INTVECTOR_215	INTVECTOR	<a href="#">Go</a>
2360h	INTVECTOR_216	INTVECTOR	<a href="#">Go</a>
2364h	INTVECTOR_217	INTVECTOR	<a href="#">Go</a>
2368h	INTVECTOR_218	INTVECTOR	<a href="#">Go</a>
236Ch	INTVECTOR_219	INTVECTOR	<a href="#">Go</a>
2370h	INTVECTOR_220	INTVECTOR	<a href="#">Go</a>
2374h	INTVECTOR_221	INTVECTOR	<a href="#">Go</a>
2378h	INTVECTOR_222	INTVECTOR	<a href="#">Go</a>
237Ch	INTVECTOR_223	INTVECTOR	<a href="#">Go</a>
2380h	INTVECTOR_224	INTVECTOR	<a href="#">Go</a>
2384h	INTVECTOR_225	INTVECTOR	<a href="#">Go</a>
2388h	INTVECTOR_226	INTVECTOR	<a href="#">Go</a>
238Ch	INTVECTOR_227	INTVECTOR	<a href="#">Go</a>
2390h	INTVECTOR_228	INTVECTOR	<a href="#">Go</a>
2394h	INTVECTOR_229	INTVECTOR	<a href="#">Go</a>
2398h	INTVECTOR_230	INTVECTOR	<a href="#">Go</a>
239Ch	INTVECTOR_231	INTVECTOR	<a href="#">Go</a>
23A0h	INTVECTOR_232	INTVECTOR	<a href="#">Go</a>
23A4h	INTVECTOR_233	INTVECTOR	<a href="#">Go</a>
23A8h	INTVECTOR_234	INTVECTOR	<a href="#">Go</a>
23ACh	INTVECTOR_235	INTVECTOR	<a href="#">Go</a>
23B0h	INTVECTOR_236	INTVECTOR	<a href="#">Go</a>
23B4h	INTVECTOR_237	INTVECTOR	<a href="#">Go</a>
23B8h	INTVECTOR_238	INTVECTOR	<a href="#">Go</a>
23BCh	INTVECTOR_239	INTVECTOR	<a href="#">Go</a>
23C0h	INTVECTOR_240	INTVECTOR	<a href="#">Go</a>
23C4h	INTVECTOR_241	INTVECTOR	<a href="#">Go</a>
23C8h	INTVECTOR_242	INTVECTOR	<a href="#">Go</a>
23CCh	INTVECTOR_243	INTVECTOR	<a href="#">Go</a>
23D0h	INTVECTOR_244	INTVECTOR	<a href="#">Go</a>
23D4h	INTVECTOR_245	INTVECTOR	<a href="#">Go</a>
23D8h	INTVECTOR_246	INTVECTOR	<a href="#">Go</a>
23DCh	INTVECTOR_247	INTVECTOR	<a href="#">Go</a>
23E0h	INTVECTOR_248	INTVECTOR	<a href="#">Go</a>
23E4h	INTVECTOR_249	INTVECTOR	<a href="#">Go</a>

**Table 13-1. MSS\_VIM Registers (continued)**

Offset	Acronym	Register Name	Section
23E8h	INTVECTOR_250	INTVECTOR	<a href="#">Go</a>
23ECh	INTVECTOR_251	INTVECTOR	<a href="#">Go</a>
23F0h	INTVECTOR_252	INTVECTOR	<a href="#">Go</a>
23F4h	INTVECTOR_253	INTVECTOR	<a href="#">Go</a>
23F8h	INTVECTOR_254	INTVECTOR	<a href="#">Go</a>
23FCh	INTVECTOR_255	INTVECTOR	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 13-2](#) shows the codes that are used for access types in this section.

**Table 13-2. MSS\_VIM Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 13.10.1 PID Register (Offset = 0h) [Reset = 00000000h]

PID is shown in [Table 13-3](#).

Return to the [Summary Table](#).

The Revision Register contains the major and minor revisions for the module.

**Table 13-3. PID Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	SCHEME	R	1h	PID register scheme
29-28	BU	R	2h	Business Unit: 10 = Processors
27-16	FUNC	R	90h	Module ID
15-11	RTL	R	0h	RTL revision. Will vary depending on release.
10-8	MAJOR	R	0h	Major revision
7-6	CUSTOM	R	0h	Custom
5-0	MINOR	R	1h	Minor revision

### 13.10.2 INFO Register (Offset = 4h) [Reset = 00000000h]

INFO is shown in [Table 13-4](#).

Return to the [Summary Table](#).

The Info Register gives the configuration Information of this VIM.

**Table 13-4. INFO Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RES1	R	0h	RESERVE FIELD
10-0	INTERRUPTS	R	100h	Total number of Interrupts

### 13.10.3 PRIIRQ Register (Offset = 8h) [Reset = 00000000h]

PRIIRQ is shown in [Table 13-5](#).

Return to the [Summary Table](#).

The Prioritized IRQ Register shows the number of the highest priority pending IRQ.

**Table 13-5. PRIIRQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	VALID	R	0h	Indicates that the num field is valid.
30-20	RES2	R	0h	RESERVE FIELD
19-16	PRI	R	0h	Priority of the highest priority pending IRQ. valid only if the valid flag is set.
15-10	RES3	R	0h	RESERVE FIELD
9-0	NUM	R	0h	Number of the highest priority pending IRQ. valid only if the valid flag is set.

### 13.10.4 PRIFIQ Register (Offset = Ch) [Reset = 00000000h]

PRIFIQ is shown in [Table 13-6](#).

Return to the [Summary Table](#).

The Prioritized FIQ Register shows the number of the highest priority pending FIQ.

**Table 13-6. PRIFIQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	VALID	R	0h	Indicates that the num field is valid.
30-20	RES4	R	0h	RESERVE FIELD
19-16	PRI	R	0h	Priority of the highest priority pending FIQ. valid only if the valid flag is set.
15-10	RES5	R	0h	RESERVE FIELD
9-0	NUM	R	0h	Number of the highest priority pending FIQ. valid only if the valid flag is set.

### 13.10.5 IRQGSTS Register (Offset = 10h) [Reset = 00000000h]

IRQGSTS is shown in [Table 13-7](#).

Return to the [Summary Table](#).

The IRQ Group Status Register indicates which groups have pending IRQ interrupts.

**Table 13-7. IRQGSTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R	0h	Indicates that the num field is valid.



### 13.10.6 FIQGSTS Register (Offset = 14h) [Reset = 00000000h]

FIQGSTS is shown in [Table 13-8](#).

Return to the [Summary Table](#).

The FIQ Group Status Register indicates which groups have pending FIQ interrupts.

**Table 13-8. FIQGSTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R	0h	Indicates that the num field is valid.

### 13.10.7 IRQVEC Register (Offset = 18h) [Reset = 00000000h]

IRQVEC is shown in [Table 13-9](#).

Return to the [Summary Table](#).

The IRQ Vector Address Register contains the 32-bit address of the interrupt vector for the current pending IRQ.

**Table 13-9. IRQVEC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	Upper 30 bits of the 32-bit vector address. Only valid if the Prioritized IRQ Register valid flag is true.
1-0	RES21	R	0h	RESERVE FIELD

### 13.10.8 FIQVEC Register (Offset = 1Ch) [Reset = 00000000h]

FIQVEC is shown in [Table 13-10](#).

Return to the [Summary Table](#).

The FIQ Vector Address Register contains the 32-bit address of the interrupt vector for the current pending FIQ.

**Table 13-10. FIQVEC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	Upper 30 bits of the 32-bit vector address. Only valid if the Prioritized FIQ Register valid flag is true.
1-0	RES22	R	0h	RESERVE FIELD

### 13.10.9 ACTIRQ Register (Offset = 20h) [Reset = 00000000h]

ACTIRQ is shown in [Table 13-11](#).

Return to the [Summary Table](#).

The Active IRQ Register shows the number of the currently active IRQ.

**Table 13-11. ACTIRQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	VALID	R	0h	Indicates that the num field is valid. Set when the IRQ Vector Address Register is read and cleared whenever the IRQ Vector Address Register is written.
30-20	RES6	R	0h	RESERVE FIELD
19-16	PRI	R	0h	Priority of the highest priority pending IRQ. valid only if the valid flag is set.
15-10	RES7	R	0h	RESERVE FIELD
9-0	NUM	R	0h	Number of the currently active IRQ. Loaded from teh Prioritized IRQ Register whenever the IRQ Vector Address is read. Valid only if the valid flag is set.

### 13.10.10 ACTFIQ Register (Offset = 24h) [Reset = 00000000h]

ACTFIQ is shown in [Table 13-12](#).

Return to the [Summary Table](#).

The Active FIQ Register shows the number of the currently active FIQ.

**Table 13-12. ACTFIQ Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	VALID	R	0h	Indicates that the num field is valid. Set when the FIQ Vector Address Register is read and cleared whenever the FIQ Vector Address Register is written.
30-20	RES8	R	0h	RESERVE FIELD
19-16	PRI	R	0h	Priority of the highest priority pending IRQ. valid only if the valid flag is set.
15-10	RES9	R	0h	RESERVE FIELD
9-0	NUM	R	0h	Number of the currently active FIQ. Loaded from teh Prioritized FIQ Register whenever the FIQ Vector Address is read. Valid only if the valid flag is set.

### 13.10.11 DEDVEC Register (Offset = 30h) [Reset = 00000000h]

DEDVEC is shown in [Table 13-13](#).

Return to the [Summary Table](#).

The DED Vector Address contains a default vector address for when an uncorrectable error is detected for an active IRQ or FIQ.

**Table 13-13. DEDVEC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	Upper 30 bits of the 32-bit vector address.
1-0	RES23	R	0h	RESERVE FIELD

### 13.10.12 RAW Register (Offset = 400h) [Reset = 00000000h]

RAW is shown in [Table 13-14](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-14. RAW Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.13 STS Register (Offset = 404h) [Reset = 00000000h]

STS is shown in [Table 13-15](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-15. STS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.14 INTR\_EN\_SET Register (Offset = 408h) [Reset = 00000000h]

INTR\_EN\_SET is shown in [Table 13-16](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-16. INTR\_EN\_SET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.15 INTER\_EN\_CLR Register (Offset = 40Ch) [Reset = 00000000h]

INTER\_EN\_CLR is shown in [Table 13-17](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-17. INTER\_EN\_CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.16 IRQSTS Register (Offset = 410h) [Reset = 00000000h]

IRQSTS is shown in [Table 13-18](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h10$

**Table 13-18. IRQSTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ . Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.17 FIQSTS Register (Offset = 414h) [Reset = 00000000h]

FIQSTS is shown in [Table 13-19](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h14$

**Table 13-19. FIQSTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.18 INTMAP Register (Offset = 418h) [Reset = 00000000h]

INTMAP is shown in [Table 13-20](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $\text{h}400 + \text{M} \times \text{h}20 + \text{h}18$

**Table 13-20. INTMAP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = \text{M} \times 32 + \text{Bit } 0$ IRQ Interrupt (default) 1 FIQ Interrupt

### 13.10.19 INTTYPE Register (Offset = 41Ch) [Reset = 00000000h]

INTTYPE is shown in [Table 13-21](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7)  $h400 + M \times h20 + 0x1C$

**Table 13-21. INTTYPE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where <math>Q = M \times 32 + \text{Bit 0 Level}</math> (default) 1 Pulse</p>

### 13.10.20 RAW\_1 Register (Offset = 420h) [Reset = 00000000h]

RAW\_1 is shown in [Table 13-22](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-22. RAW\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.21 STS\_1 Register (Offset = 424h) [Reset = 00000000h]

STS\_1 is shown in [Table 13-23](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-23. STS\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.22 INTR\_EN\_SET\_1 Register (Offset = 428h) [Reset = 00000000h]

INTR\_EN\_SET\_1 is shown in [Table 13-24](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-24. INTR\_EN\_SET\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.23 INTER\_EN\_CLR\_1 Register (Offset = 42Ch) [Reset = 00000000h]

INTER\_EN\_CLR\_1 is shown in [Table 13-25](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-25. INTER\_EN\_CLR\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.24 IRQSTS\_1 Register (Offset = 430h) [Reset = 00000000h]

IRQSTS\_1 is shown in [Table 13-26](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h10$

**Table 13-26. IRQSTS\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ . Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.25 FIQSTS\_1 Register (Offset = 434h) [Reset = 00000000h]

FIQSTS\_1 is shown in [Table 13-27](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-27. FIQSTS\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.26 INTMAP\_1 Register (Offset = 438h) [Reset = 00000000h]

INTMAP\_1 is shown in [Table 13-28](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $h400 + M \times h20 + h18$

**Table 13-28. INTMAP\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit 0}$ IRQ Interrupt (default) 1 FIQ Interrupt

### 13.10.27 INTTYPE\_1 Register (Offset = 43Ch) [Reset = 00000000h]

INTTYPE\_1 is shown in [Table 13-29](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-29. INTTYPE\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.28 RAW\_2 Register (Offset = 440h) [Reset = 00000000h]

RAW\_2 is shown in [Table 13-30](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-30. RAW\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.29 STS\_2 Register (Offset = 444h) [Reset = 00000000h]

STS\_2 is shown in [Table 13-31](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-31. STS\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.30 INTR\_EN\_SET\_2 Register (Offset = 448h) [Reset = 00000000h]

INTR\_EN\_SET\_2 is shown in [Table 13-32](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-32. INTR\_EN\_SET\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.31 INTER\_EN\_CLR\_2 Register (Offset = 44Ch) [Reset = 00000000h]

INTER\_EN\_CLR\_2 is shown in [Table 13-33](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-33. INTER\_EN\_CLR\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.32 IRQSTS\_2 Register (Offset = 450h) [Reset = 00000000h]

IRQSTS\_2 is shown in [Table 13-34](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h10$

**Table 13-34. IRQSTS\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ . Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.33 FIQSTS\_2 Register (Offset = 454h) [Reset = 00000000h]

FIQSTS\_2 is shown in [Table 13-35](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-35. FIQSTS\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.34 INTMAP\_2 Register (Offset = 458h) [Reset = 00000000h]

INTMAP\_2 is shown in [Table 13-36](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $h400 + M \times h20 + h18$

**Table 13-36. INTMAP\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit 0}$ IRQ Interrupt (default) 1 FIQ Interrupt

### 13.10.35 INTTYPE\_2 Register (Offset = 45Ch) [Reset = 00000000h]

INTTYPE\_2 is shown in [Table 13-37](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-37. INTTYPE\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.36 RAW\_3 Register (Offset = 460h) [Reset = 00000000h]

RAW\_3 is shown in [Table 13-38](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-38. RAW\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.37 STS\_3 Register (Offset = 464h) [Reset = 00000000h]

STS\_3 is shown in [Table 13-39](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-39. STS\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.38 INTR\_EN\_SET\_3 Register (Offset = 468h) [Reset = 00000000h]

INTR\_EN\_SET\_3 is shown in [Table 13-40](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-40. INTR\_EN\_SET\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.39 INTER\_EN\_CLR\_3 Register (Offset = 46Ch) [Reset = 00000000h]

INTER\_EN\_CLR\_3 is shown in [Table 13-41](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-41. INTER\_EN\_CLR\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.40 IRQSTS\_3 Register (Offset = 470h) [Reset = 00000000h]

IRQSTS\_3 is shown in [Table 13-42](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h10$

**Table 13-42. IRQSTS\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ . Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.41 FIQSTS\_3 Register (Offset = 474h) [Reset = 00000000h]

FIQSTS\_3 is shown in [Table 13-43](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-43. FIQSTS\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.42 INTMAP\_3 Register (Offset = 478h) [Reset = 00000000h]

INTMAP\_3 is shown in [Table 13-44](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $\text{h400} + \text{M} \times \text{h20} + \text{h18}$

**Table 13-44. INTMAP\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = \text{Mx32} + \text{Bit 0 IRQ Interrupt}$ (default) 1 FIQ Interrupt

### 13.10.43 INTTYPE\_3 Register (Offset = 47Ch) [Reset = 00000000h]

INTTYPE\_3 is shown in [Table 13-45](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-45. INTTYPE\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.44 RAW\_4 Register (Offset = 480h) [Reset = 00000000h]

RAW\_4 is shown in [Table 13-46](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-46. RAW\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.45 STS\_4 Register (Offset = 484h) [Reset = 00000000h]

STS\_4 is shown in [Table 13-47](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-47. STS\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.46 INTR\_EN\_SET\_4 Register (Offset = 488h) [Reset = 00000000h]

INTR\_EN\_SET\_4 is shown in [Table 13-48](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-48. INTR\_EN\_SET\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.47 INTER\_EN\_CLR\_4 Register (Offset = 48Ch) [Reset = 00000000h]

INTER\_EN\_CLR\_4 is shown in [Table 13-49](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-49. INTER\_EN\_CLR\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.48 IRQSTS\_4 Register (Offset = 490h) [Reset = 00000000h]

IRQSTS\_4 is shown in [Table 13-50](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h10$

**Table 13-50. IRQSTS\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ . Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.49 FIQSTS\_4 Register (Offset = 494h) [Reset = 00000000h]

FIQSTS\_4 is shown in [Table 13-51](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-51. FIQSTS\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.50 INTMAP\_4 Register (Offset = 498h) [Reset = 00000000h]

INTMAP\_4 is shown in [Table 13-52](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $\text{h}400 + \text{M} \times \text{h}20 + \text{h}18$

**Table 13-52. INTMAP\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = \text{M} \times 32 + \text{Bit } 0$ IRQ Interrupt (default) 1 FIQ Interrupt

### 13.10.51 INTTYPE\_4 Register (Offset = 49Ch) [Reset = 00000000h]

INTTYPE\_4 is shown in [Table 13-53](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-53. INTTYPE\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.52 RAW\_5 Register (Offset = 4A0h) [Reset = 00000000h]

RAW\_5 is shown in [Table 13-54](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-54. RAW\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.53 STS\_5 Register (Offset = 4A4h) [Reset = 00000000h]

STS\_5 is shown in [Table 13-55](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h04

**Table 13-55. STS\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.54 INTR\_EN\_SET\_5 Register (Offset = 4A8h) [Reset = 00000000h]

INTR\_EN\_SET\_5 is shown in [Table 13-56](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7) h400 + M x h20 + h08

**Table 13-56. INTR\_EN\_SET\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.55 INTER\_EN\_CLR\_5 Register (Offset = 4ACh) [Reset = 00000000h]

INTER\_EN\_CLR\_5 is shown in [Table 13-57](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-57. INTER\_EN\_CLR\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.56 IRQSTS\_5 Register (Offset = 4B0h) [Reset = 00000000h]

IRQSTS\_5 is shown in [Table 13-58](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h10

**Table 13-58. IRQSTS\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where Q = Mx32+Bit. Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.57 FIQSTS\_5 Register (Offset = 4B4h) [Reset = 00000000h]

FIQSTS\_5 is shown in [Table 13-59](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-59. FIQSTS\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.58 INTMAP\_5 Register (Offset = 4B8h) [Reset = 00000000h]

INTMAP\_5 is shown in [Table 13-60](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $h400 + M \times h20 + h18$

**Table 13-60. INTMAP\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit 0}$ IRQ Interrupt (default) 1 FIQ Interrupt

### 13.10.59 INTTYPE\_5 Register (Offset = 4BCh) [Reset = 00000000h]

INTTYPE\_5 is shown in [Table 13-61](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-61. INTTYPE\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.60 RAW\_6 Register (Offset = 4C0h) [Reset = 00000000h]

RAW\_6 is shown in [Table 13-62](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-62. RAW\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.61 STS\_6 Register (Offset = 4C4h) [Reset = 00000000h]

STS\_6 is shown in [Table 13-63](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-63. STS\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.62 INTR\_EN\_SET\_6 Register (Offset = 4C8h) [Reset = 00000000h]

INTR\_EN\_SET\_6 is shown in [Table 13-64](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-64. INTR\_EN\_SET\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.63 INTER\_EN\_CLR\_6 Register (Offset = 4CCh) [Reset = 00000000h]

INTER\_EN\_CLR\_6 is shown in [Table 13-65](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-65. INTER\_EN\_CLR\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.64 IRQSTS\_6 Register (Offset = 4D0h) [Reset = 00000000h]

IRQSTS\_6 is shown in [Table 13-66](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h10

**Table 13-66. IRQSTS\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where Q = Mx32+Bit. Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.65 FIQSTS\_6 Register (Offset = 4D4h) [Reset = 00000000h]

FIQSTS\_6 is shown in [Table 13-67](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-67. FIQSTS\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.66 INTMAP\_6 Register (Offset = 4D8h) [Reset = 00000000h]

INTMAP\_6 is shown in [Table 13-68](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $h400 + M \times h20 + h18$

**Table 13-68. INTMAP\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit 0 IRQ Interrupt (default) 1 FIQ Interrupt}$

### 13.10.67 INTTYPE\_6 Register (Offset = 4DCh) [Reset = 00000000h]

INTTYPE\_6 is shown in [Table 13-69](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-69. INTTYPE\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.68 RAW\_7 Register (Offset = 4E0h) [Reset = 00000000h]

RAW\_7 is shown in [Table 13-70](#).

Return to the [Summary Table](#).

Group M Interrupt Raw Status/Set Register (M is 0 to 7)  $h400 + M \times h20 + h00$

**Table 13-70. RAW\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STS	R/W	0h	This is the raw status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive Read 1 Active/ Pending Write 0 No effect Write 1 Set to Interrupt Raw Status

### 13.10.69 STS\_7 Register (Offset = 4E4h) [Reset = 00000000h]

STS\_7 is shown in [Table 13-71](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Status/Clear Register (M is 0 to 7)  $h400 + M \times h20 + h04$

**Table 13-71. STS\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Inactive or Disabled Read 1 Active/Pending and Enabled Write 0 No effect Write 1 Clear Interrupt Raw Status



### 13.10.70 INTR\_EN\_SET\_7 Register (Offset = 4E8h) [Reset = 00000000h]

INTR\_EN\_SET\_7 is shown in [Table 13-72](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Set Register (M is 0 to 7)  $h400 + M \times h20 + h08$

**Table 13-72. INTR\_EN\_SET\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to enable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Set Enable

### 13.10.71 INTER\_EN\_CLR\_7 Register (Offset = 4ECh) [Reset = 00000000h]

INTER\_EN\_CLR\_7 is shown in [Table 13-73](#).

Return to the [Summary Table](#).

Group M Interrupt Enabled Clear Register (M is 0 to 7)  $h400 + M \times h20 + h0C$

**Table 13-73. INTER\_EN\_CLR\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to disable the mask of events in Group M Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit}$ Read 0 Disabled Read 1 Enabled Write 0 No effect Write 1 Clear Enable

### 13.10.72 IRQSTS\_7 Register (Offset = 4F0h) [Reset = 00000000h]

IRQSTS\_7 is shown in [Table 13-74](#).

Return to the [Summary Table](#).

Group M Interrupt IRQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h10

**Table 13-74. IRQSTS\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to IRQ. Each bit corresponds to event Q where Q = Mx32+Bit. Read 0 Inactive, Disabled, or not an IRQ. Read 1 Active/Pending, Enabled, and IRQ. Write 0 No effect. Write 1 Clear Interrupt Raw Status (if IRQ).

### 13.10.73 FIQSTS\_7 Register (Offset = 4F4h) [Reset = 00000000h]

FIQSTS\_7 is shown in [Table 13-75](#).

Return to the [Summary Table](#).

Group M Interrupt FIQ Enabled Status/Clear Register (M is 0 to 7) h400 + M x h20 + h14

**Table 13-75. FIQSTS\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This is the masked status of the events in group M that are mapped to FIQ Each bit corresponds to event Q where Q = Mx32+Bit Read 0 Inactive, Disabled, or not an FIQ Read 1 Active/Pending, Enabled, and FIQ Write 0 No effect Write 1 Clear Interrupt Raw Status (if FIQ)

### 13.10.74 INTMAP\_7 Register (Offset = 4F8h) [Reset = 00000000h]

INTMAP\_7 is shown in [Table 13-76](#).

Return to the [Summary Table](#).

Group M Interrupt Map Register (M is 0 to 7)  $h400 + M \times h20 + h18$

**Table 13-76. INTMAP\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK	R/W	0h	This field is used to indicate which interrupt the corresponding event influences (if enabled) for event group M. Each bit corresponds to event Q where $Q = M \times 32 + \text{Bit 0}$ IRQ Interrupt (default) 1 FIQ Interrupt

### 13.10.75 INTTYPE\_7 Register (Offset = 4FCh) [Reset = 00000000h]

INTTYPE\_7 is shown in [Table 13-77](#).

Return to the [Summary Table](#).

Group M Type Map Register (M is 0 to 7) h400 + M x h20 + 0x1C

**Table 13-77. INTTYPE\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	VAL	R/W	0h	<p>This field is used to indicate whether the source of an interrupt is a level (default) or a pulse for event group M.</p> <p>This is informational so that an ISR may query this register and know whether it has to clear a pulse event or a level event (see 3.4 Interrupt Handling).</p> <p>The value has no effect on how the VIM hardware functions.</p> <p>The input interrupts are agnostic as to whether they are pulse or level.</p> <p>Each bit corresponds to event Q where Q = Mx32+Bit 0 Level (default) 1 Pulse</p>

### 13.10.76 INTPRIORITY Register (Offset = 1000h) [Reset = 00000000h]

INTPRIORITY is shown in [Table 13-78](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h4

**Table 13-78. INTPRIORITY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.77 INTPRIORITY\_1 Register (Offset = 1004h) [Reset = 00000000h]

INTPRIORITY\_1 is shown in [Table 13-79](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h5

**Table 13-79. INTPRIORITY\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.78 INTPRIORITY\_2 Register (Offset = 1008h) [Reset = 00000000h]

INTPRIORITY\_2 is shown in [Table 13-80](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h6

**Table 13-80. INTPRIORITY\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.79 INTPRIORITY\_3 Register (Offset = 100Ch) [Reset = 00000000h]

INTPRIORITY\_3 is shown in [Table 13-81](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h7

**Table 13-81. INTPRIORITY\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.80 INTPRIORITY\_4 Register (Offset = 1010h) [Reset = 00000000h]

INTPRIORITY\_4 is shown in [Table 13-82](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h8

**Table 13-82. INTPRIORITY\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.81 INTPRIORITY\_5 Register (Offset = 1014h) [Reset = 00000000h]

INTPRIORITY\_5 is shown in [Table 13-83](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h9

**Table 13-83. INTPRIORITY\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.82 INTPRIORITY\_6 Register (Offset = 1018h) [Reset = 00000000h]

INTPRIORITY\_6 is shown in [Table 13-84](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h10

**Table 13-84. INTPRIORITY\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.83 INTPRIORITY\_7 Register (Offset = 101Ch) [Reset = 00000000h]

INTPRIORITY\_7 is shown in [Table 13-85](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h11

**Table 13-85. INTPRIORITY\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.84 INTPRIORITY\_8 Register (Offset = 1020h) [Reset = 00000000h]

INTPRIORITY\_8 is shown in [Table 13-86](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h12

**Table 13-86. INTPRIORITY\_8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.85 INTPRIORITY\_9 Register (Offset = 1024h) [Reset = 00000000h]

INTPRIORITY\_9 is shown in [Table 13-87](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h13

**Table 13-87. INTPRIORITY\_9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.86 INTPRIORITY\_10 Register (Offset = 1028h) [Reset = 00000000h]

INTPRIORITY\_10 is shown in [Table 13-88](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h14

**Table 13-88. INTPRIORITY\_10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.87 INTPRIORITY\_11 Register (Offset = 102Ch) [Reset = 00000000h]

INTPRIORITY\_11 is shown in [Table 13-89](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h15

**Table 13-89. INTPRIORITY\_11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.88 INTPRIORITY\_12 Register (Offset = 1030h) [Reset = 00000000h]

INTPRIORITY\_12 is shown in [Table 13-90](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h16

**Table 13-90. INTPRIORITY\_12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.89 INTPRIORITY\_13 Register (Offset = 1034h) [Reset = 00000000h]

INTPRIORITY\_13 is shown in [Table 13-91](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h17

**Table 13-91. INTPRIORITY\_13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.90 INTPRIORITY\_14 Register (Offset = 1038h) [Reset = 00000000h]

INTPRIORITY\_14 is shown in [Table 13-92](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h18

**Table 13-92. INTPRIORITY\_14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.91 INTPRIORITY\_15 Register (Offset = 103Ch) [Reset = 00000000h]

INTPRIORITY\_15 is shown in [Table 13-93](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h19

**Table 13-93. INTPRIORITY\_15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.92 INTPRIORITY\_16 Register (Offset = 1040h) [Reset = 00000000h]

INTPRIORITY\_16 is shown in [Table 13-94](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h20

**Table 13-94. INTPRIORITY\_16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.93 INTPRIORITY\_17 Register (Offset = 1044h) [Reset = 00000000h]

INTPRIORITY\_17 is shown in [Table 13-95](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h21

**Table 13-95. INTPRIORITY\_17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.94 INTPRIORITY\_18 Register (Offset = 1048h) [Reset = 00000000h]

INTPRIORITY\_18 is shown in [Table 13-96](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h22

**Table 13-96. INTPRIORITY\_18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.95 INTPRIORITY\_19 Register (Offset = 104Ch) [Reset = 00000000h]

INTPRIORITY\_19 is shown in [Table 13-97](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h23

**Table 13-97. INTPRIORITY\_19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.96 INTPRIORITY\_20 Register (Offset = 1050h) [Reset = 00000000h]

INTPRIORITY\_20 is shown in [Table 13-98](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h24

**Table 13-98. INTPRIORITY\_20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.97 INTPRIORITY\_21 Register (Offset = 1054h) [Reset = 00000000h]

INTPRIORITY\_21 is shown in [Table 13-99](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h25

**Table 13-99. INTPRIORITY\_21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.98 INTPRIORITY\_22 Register (Offset = 1058h) [Reset = 00000000h]

INTPRIORITY\_22 is shown in [Table 13-100](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h26

**Table 13-100. INTPRIORITY\_22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.99 INTPRIORITY\_23 Register (Offset = 105Ch) [Reset = 00000000h]

INTPRIORITY\_23 is shown in [Table 13-101](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h27

**Table 13-101. INTPRIORITY\_23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.100 INTPRIORITY\_24 Register (Offset = 1060h) [Reset = 00000000h]

INTPRIORITY\_24 is shown in [Table 13-102](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h28

**Table 13-102. INTPRIORITY\_24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.101 INTPRIORITY\_25 Register (Offset = 1064h) [Reset = 00000000h]

INTPRIORITY\_25 is shown in [Table 13-103](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h29

**Table 13-103. INTPRIORITY\_25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.102 INTPRIORITY\_26 Register (Offset = 1068h) [Reset = 00000000h]

INTPRIORITY\_26 is shown in [Table 13-104](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h30

**Table 13-104. INTPRIORITY\_26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.103 INTPRIORITY\_27 Register (Offset = 106Ch) [Reset = 00000000h]

INTPRIORITY\_27 is shown in [Table 13-105](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h31

**Table 13-105. INTPRIORITY\_27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.104 INTPRIORITY\_28 Register (Offset = 1070h) [Reset = 00000000h]

INTPRIORITY\_28 is shown in [Table 13-106](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h32

**Table 13-106. INTPRIORITY\_28 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.105 INTPRIORITY\_29 Register (Offset = 1074h) [Reset = 00000000h]

INTPRIORITY\_29 is shown in [Table 13-107](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h33

**Table 13-107. INTPRIORITY\_29 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.106 INTPRIORITY\_30 Register (Offset = 1078h) [Reset = 00000000h]

INTPRIORITY\_30 is shown in [Table 13-108](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h34

**Table 13-108. INTPRIORITY\_30 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.107 INTPRIORITY\_31 Register (Offset = 107Ch) [Reset = 00000000h]

INTPRIORITY\_31 is shown in [Table 13-109](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h35

**Table 13-109. INTPRIORITY\_31 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.108 INTPRIORITY\_32 Register (Offset = 1080h) [Reset = 00000000h]

INTPRIORITY\_32 is shown in [Table 13-110](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h36

**Table 13-110. INTPRIORITY\_32 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.109 INTPRIORITY\_33 Register (Offset = 1084h) [Reset = 00000000h]

INTPRIORITY\_33 is shown in [Table 13-111](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h37

**Table 13-111. INTPRIORITY\_33 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.110 INTPRIORITY\_34 Register (Offset = 1088h) [Reset = 00000000h]

INTPRIORITY\_34 is shown in [Table 13-112](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h38

**Table 13-112. INTPRIORITY\_34 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.111 INTPRIORITY\_35 Register (Offset = 108Ch) [Reset = 00000000h]

INTPRIORITY\_35 is shown in [Table 13-113](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h39

**Table 13-113. INTPRIORITY\_35 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.112 INTPRIORITY\_36 Register (Offset = 1090h) [Reset = 00000000h]

INTPRIORITY\_36 is shown in [Table 13-114](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h40

**Table 13-114. INTPRIORITY\_36 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.113 INTPRIORITY\_37 Register (Offset = 1094h) [Reset = 00000000h]

INTPRIORITY\_37 is shown in [Table 13-115](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h41

**Table 13-115. INTPRIORITY\_37 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.114 INTPRIORITY\_38 Register (Offset = 1098h) [Reset = 00000000h]

INTPRIORITY\_38 is shown in [Table 13-116](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h42

**Table 13-116. INTPRIORITY\_38 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.115 INTPRIORITY\_39 Register (Offset = 109Ch) [Reset = 00000000h]

INTPRIORITY\_39 is shown in [Table 13-117](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h43

**Table 13-117. INTPRIORITY\_39 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.116 INTPRIORITY\_40 Register (Offset = 10A0h) [Reset = 00000000h]

INTPRIORITY\_40 is shown in [Table 13-118](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h44

**Table 13-118. INTPRIORITY\_40 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.117 INTPRIORITY\_41 Register (Offset = 10A4h) [Reset = 00000000h]

INTPRIORITY\_41 is shown in [Table 13-119](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h45

**Table 13-119. INTPRIORITY\_41 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.118 INTPRIORITY\_42 Register (Offset = 10A8h) [Reset = 00000000h]

INTPRIORITY\_42 is shown in [Table 13-120](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h46

**Table 13-120. INTPRIORITY\_42 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.119 INTPRIORITY\_43 Register (Offset = 10ACh) [Reset = 00000000h]

INTPRIORITY\_43 is shown in [Table 13-121](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h47

**Table 13-121. INTPRIORITY\_43 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.120 INTPRIORITY\_44 Register (Offset = 10B0h) [Reset = 00000000h]

INTPRIORITY\_44 is shown in [Table 13-122](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h48

**Table 13-122. INTPRIORITY\_44 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.121 INTPRIORITY\_45 Register (Offset = 10B4h) [Reset = 00000000h]

INTPRIORITY\_45 is shown in [Table 13-123](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h49

**Table 13-123. INTPRIORITY\_45 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.122 INTPRIORITY\_46 Register (Offset = 10B8h) [Reset = 00000000h]

INTPRIORITY\_46 is shown in [Table 13-124](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h50

**Table 13-124. INTPRIORITY\_46 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.123 INTPRIORITY\_47 Register (Offset = 10BCh) [Reset = 00000000h]

INTPRIORITY\_47 is shown in [Table 13-125](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h51

**Table 13-125. INTPRIORITY\_47 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.124 INTPRIORITY\_48 Register (Offset = 10C0h) [Reset = 00000000h]

INTPRIORITY\_48 is shown in [Table 13-126](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h52

**Table 13-126. INTPRIORITY\_48 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.125 INTPRIORITY\_49 Register (Offset = 10C4h) [Reset = 00000000h]

INTPRIORITY\_49 is shown in [Table 13-127](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h53

**Table 13-127. INTPRIORITY\_49 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.126 INTPRIORITY\_50 Register (Offset = 10C8h) [Reset = 00000000h]

INTPRIORITY\_50 is shown in [Table 13-128](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h54

**Table 13-128. INTPRIORITY\_50 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.127 INTPRIORITY\_51 Register (Offset = 10CCh) [Reset = 00000000h]

INTPRIORITY\_51 is shown in [Table 13-129](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h55

**Table 13-129. INTPRIORITY\_51 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.128 INTPRIORITY\_52 Register (Offset = 10D0h) [Reset = 00000000h]

INTPRIORITY\_52 is shown in [Table 13-130](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h56

**Table 13-130. INTPRIORITY\_52 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.129 INTPRIORITY\_53 Register (Offset = 10D4h) [Reset = 00000000h]

INTPRIORITY\_53 is shown in [Table 13-131](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h57

**Table 13-131. INTPRIORITY\_53 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.130 INTPRIORITY\_54 Register (Offset = 10D8h) [Reset = 00000000h]

INTPRIORITY\_54 is shown in [Table 13-132](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h58

**Table 13-132. INTPRIORITY\_54 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.131 INTPRIORITY\_55 Register (Offset = 10DCh) [Reset = 00000000h]

INTPRIORITY\_55 is shown in [Table 13-133](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h59

**Table 13-133. INTPRIORITY\_55 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.132 INTPRIORITY\_56 Register (Offset = 10E0h) [Reset = 00000000h]

INTPRIORITY\_56 is shown in [Table 13-134](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h60

**Table 13-134. INTPRIORITY\_56 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.133 INTPRIORITY\_57 Register (Offset = 10E4h) [Reset = 00000000h]

INTPRIORITY\_57 is shown in [Table 13-135](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h61

**Table 13-135. INTPRIORITY\_57 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.134 INTPRIORITY\_58 Register (Offset = 10E8h) [Reset = 00000000h]

INTPRIORITY\_58 is shown in [Table 13-136](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h62

**Table 13-136. INTPRIORITY\_58 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.135 INTPRIORITY\_59 Register (Offset = 10ECh) [Reset = 00000000h]

INTPRIORITY\_59 is shown in [Table 13-137](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h63

**Table 13-137. INTPRIORITY\_59 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.136 INTPRIORITY\_60 Register (Offset = 10F0h) [Reset = 00000000h]

INTPRIORITY\_60 is shown in [Table 13-138](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h64

**Table 13-138. INTPRIORITY\_60 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.137 INTPRIORITY\_61 Register (Offset = 10F4h) [Reset = 00000000h]

INTPRIORITY\_61 is shown in [Table 13-139](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h65

**Table 13-139. INTPRIORITY\_61 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.138 INTPRIORITY\_62 Register (Offset = 10F8h) [Reset = 00000000h]

INTPRIORITY\_62 is shown in [Table 13-140](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h66

**Table 13-140. INTPRIORITY\_62 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.139 INTPRIORITY\_63 Register (Offset = 10FCh) [Reset = 00000000h]

INTPRIORITY\_63 is shown in [Table 13-141](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h67

**Table 13-141. INTPRIORITY\_63 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.140 INTPRIORITY\_64 Register (Offset = 1100h) [Reset = 00000000h]

INTPRIORITY\_64 is shown in [Table 13-142](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h68

**Table 13-142. INTPRIORITY\_64 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.141 INTPRIORITY\_65 Register (Offset = 1104h) [Reset = 00000000h]

INTPRIORITY\_65 is shown in [Table 13-143](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h69

**Table 13-143. INTPRIORITY\_65 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.142 INTPRIORITY\_66 Register (Offset = 1108h) [Reset = 00000000h]

INTPRIORITY\_66 is shown in [Table 13-144](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h70

**Table 13-144. INTPRIORITY\_66 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.143 INTPRIORITY\_67 Register (Offset = 110Ch) [Reset = 00000000h]

INTPRIORITY\_67 is shown in [Table 13-145](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h71

**Table 13-145. INTPRIORITY\_67 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.144 INTPRIORITY\_68 Register (Offset = 1110h) [Reset = 00000000h]

INTPRIORITY\_68 is shown in [Table 13-146](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h72

**Table 13-146. INTPRIORITY\_68 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.145 INTPRIORITY\_69 Register (Offset = 1114h) [Reset = 00000000h]

INTPRIORITY\_69 is shown in [Table 13-147](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h73

**Table 13-147. INTPRIORITY\_69 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.146 INTPRIORITY\_70 Register (Offset = 1118h) [Reset = 00000000h]

INTPRIORITY\_70 is shown in [Table 13-148](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h74

**Table 13-148. INTPRIORITY\_70 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.147 INTPRIORITY\_71 Register (Offset = 111Ch) [Reset = 00000000h]

INTPRIORITY\_71 is shown in [Table 13-149](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h75

**Table 13-149. INTPRIORITY\_71 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.148 INTPRIORITY\_72 Register (Offset = 1120h) [Reset = 00000000h]

INTPRIORITY\_72 is shown in [Table 13-150](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h76

**Table 13-150. INTPRIORITY\_72 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.149 INTPRIORITY\_73 Register (Offset = 1124h) [Reset = 00000000h]

INTPRIORITY\_73 is shown in [Table 13-151](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h77

**Table 13-151. INTPRIORITY\_73 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.150 INTPRIORITY\_74 Register (Offset = 1128h) [Reset = 00000000h]

INTPRIORITY\_74 is shown in [Table 13-152](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h78

**Table 13-152. INTPRIORITY\_74 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.151 INTPRIORITY\_75 Register (Offset = 112Ch) [Reset = 00000000h]

INTPRIORITY\_75 is shown in [Table 13-153](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h79

**Table 13-153. INTPRIORITY\_75 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.152 INTPRIORITY\_76 Register (Offset = 1130h) [Reset = 00000000h]

INTPRIORITY\_76 is shown in [Table 13-154](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h80

**Table 13-154. INTPRIORITY\_76 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.153 INTPRIORITY\_77 Register (Offset = 1134h) [Reset = 00000000h]

INTPRIORITY\_77 is shown in [Table 13-155](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h81

**Table 13-155. INTPRIORITY\_77 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.154 INTPRIORITY\_78 Register (Offset = 1138h) [Reset = 00000000h]

INTPRIORITY\_78 is shown in [Table 13-156](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h82

**Table 13-156. INTPRIORITY\_78 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.155 INTPRIORITY\_79 Register (Offset = 113Ch) [Reset = 00000000h]

INTPRIORITY\_79 is shown in [Table 13-157](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h83

**Table 13-157. INTPRIORITY\_79 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.156 INTPRIORITY\_80 Register (Offset = 1140h) [Reset = 00000000h]

INTPRIORITY\_80 is shown in [Table 13-158](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h84

**Table 13-158. INTPRIORITY\_80 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.157 INTPRIORITY\_81 Register (Offset = 1144h) [Reset = 00000000h]

INTPRIORITY\_81 is shown in [Table 13-159](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h85

**Table 13-159. INTPRIORITY\_81 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.158 INTPRIORITY\_82 Register (Offset = 1148h) [Reset = 00000000h]

INTPRIORITY\_82 is shown in [Table 13-160](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h86

**Table 13-160. INTPRIORITY\_82 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.159 INTPRIORITY\_83 Register (Offset = 114Ch) [Reset = 00000000h]

INTPRIORITY\_83 is shown in [Table 13-161](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h87

**Table 13-161. INTPRIORITY\_83 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.160 INTPRIORITY\_84 Register (Offset = 1150h) [Reset = 00000000h]

INTPRIORITY\_84 is shown in [Table 13-162](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h88

**Table 13-162. INTPRIORITY\_84 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.161 INTPRIORITY\_85 Register (Offset = 1154h) [Reset = 00000000h]

INTPRIORITY\_85 is shown in [Table 13-163](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h89

**Table 13-163. INTPRIORITY\_85 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.162 INTPRIORITY\_86 Register (Offset = 1158h) [Reset = 00000000h]

INTPRIORITY\_86 is shown in [Table 13-164](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h90

**Table 13-164. INTPRIORITY\_86 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.163 INTPRIORITY\_87 Register (Offset = 115Ch) [Reset = 00000000h]

INTPRIORITY\_87 is shown in [Table 13-165](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h91

**Table 13-165. INTPRIORITY\_87 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.164 INTPRIORITY\_88 Register (Offset = 1160h) [Reset = 00000000h]

INTPRIORITY\_88 is shown in [Table 13-166](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h92

**Table 13-166. INTPRIORITY\_88 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.165 INTPRIORITY\_89 Register (Offset = 1164h) [Reset = 00000000h]

INTPRIORITY\_89 is shown in [Table 13-167](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h93

**Table 13-167. INTPRIORITY\_89 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.166 INTPRIORITY\_90 Register (Offset = 1168h) [Reset = 00000000h]

INTPRIORITY\_90 is shown in [Table 13-168](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h94

**Table 13-168. INTPRIORITY\_90 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.167 INTPRIORITY\_91 Register (Offset = 116Ch) [Reset = 00000000h]

INTPRIORITY\_91 is shown in [Table 13-169](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h95

**Table 13-169. INTPRIORITY\_91 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.168 INTPRIORITY\_92 Register (Offset = 1170h) [Reset = 00000000h]

INTPRIORITY\_92 is shown in [Table 13-170](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h96

**Table 13-170. INTPRIORITY\_92 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.169 INTPRIORITY\_93 Register (Offset = 1174h) [Reset = 00000000h]

INTPRIORITY\_93 is shown in [Table 13-171](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h97

**Table 13-171. INTPRIORITY\_93 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.170 INTPRIORITY\_94 Register (Offset = 1178h) [Reset = 00000000h]

INTPRIORITY\_94 is shown in [Table 13-172](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h98

**Table 13-172. INTPRIORITY\_94 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.171 INTPRIORITY\_95 Register (Offset = 117Ch) [Reset = 00000000h]

INTPRIORITY\_95 is shown in [Table 13-173](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h99

**Table 13-173. INTPRIORITY\_95 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.172 INTPRIORITY\_96 Register (Offset = 1180h) [Reset = 00000000h]

INTPRIORITY\_96 is shown in [Table 13-174](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h100

**Table 13-174. INTPRIORITY\_96 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.173 INTPRIORITY\_97 Register (Offset = 1184h) [Reset = 00000000h]

INTPRIORITY\_97 is shown in [Table 13-175](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h101

**Table 13-175. INTPRIORITY\_97 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.174 INTPRIORITY\_98 Register (Offset = 1188h) [Reset = 00000000h]

INTPRIORITY\_98 is shown in [Table 13-176](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h102

**Table 13-176. INTPRIORITY\_98 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.175 INTPRIORITY\_99 Register (Offset = 118Ch) [Reset = 00000000h]

INTPRIORITY\_99 is shown in [Table 13-177](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h103

**Table 13-177. INTPRIORITY\_99 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.176 INTPRIORITY\_100 Register (Offset = 1190h) [Reset = 00000000h]

INTPRIORITY\_100 is shown in [Table 13-178](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h104

**Table 13-178. INTPRIORITY\_100 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.177 INTPRIORITY\_101 Register (Offset = 1194h) [Reset = 00000000h]

INTPRIORITY\_101 is shown in [Table 13-179](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h105

**Table 13-179. INTPRIORITY\_101 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.178 INTPRIORITY\_102 Register (Offset = 1198h) [Reset = 00000000h]

INTPRIORITY\_102 is shown in [Table 13-180](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h106

**Table 13-180. INTPRIORITY\_102 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.179 INTPRIORITY\_103 Register (Offset = 119Ch) [Reset = 00000000h]

INTPRIORITY\_103 is shown in [Table 13-181](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h107

**Table 13-181. INTPRIORITY\_103 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.180 INTPRIORITY\_104 Register (Offset = 11A0h) [Reset = 00000000h]

INTPRIORITY\_104 is shown in [Table 13-182](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h108

**Table 13-182. INTPRIORITY\_104 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.181 INTPRIORITY\_105 Register (Offset = 11A4h) [Reset = 00000000h]

INTPRIORITY\_105 is shown in [Table 13-183](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h109

**Table 13-183. INTPRIORITY\_105 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.182 INTPRIORITY\_106 Register (Offset = 11A8h) [Reset = 00000000h]

INTPRIORITY\_106 is shown in [Table 13-184](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h110

**Table 13-184. INTPRIORITY\_106 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.183 INTPRIORITY\_107 Register (Offset = 11ACh) [Reset = 00000000h]

INTPRIORITY\_107 is shown in [Table 13-185](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h111

**Table 13-185. INTPRIORITY\_107 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.184 INTPRIORITY\_108 Register (Offset = 11B0h) [Reset = 00000000h]

INTPRIORITY\_108 is shown in [Table 13-186](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h112

**Table 13-186. INTPRIORITY\_108 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.185 INTPRIORITY\_109 Register (Offset = 11B4h) [Reset = 00000000h]

INTPRIORITY\_109 is shown in [Table 13-187](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h113

**Table 13-187. INTPRIORITY\_109 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.186 INTPRIORITY\_110 Register (Offset = 11B8h) [Reset = 00000000h]

INTPRIORITY\_110 is shown in [Table 13-188](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h114

**Table 13-188. INTPRIORITY\_110 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.187 INTPRIORITY\_111 Register (Offset = 11BCh) [Reset = 00000000h]

INTPRIORITY\_111 is shown in [Table 13-189](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h115

**Table 13-189. INTPRIORITY\_111 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.188 INTPRIORITY\_112 Register (Offset = 11C0h) [Reset = 00000000h]

INTPRIORITY\_112 is shown in [Table 13-190](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h116

**Table 13-190. INTPRIORITY\_112 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.189 INTPRIORITY\_113 Register (Offset = 11C4h) [Reset = 00000000h]

INTPRIORITY\_113 is shown in [Table 13-191](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h117

**Table 13-191. INTPRIORITY\_113 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.190 INTPRIORITY\_114 Register (Offset = 11C8h) [Reset = 00000000h]

INTPRIORITY\_114 is shown in [Table 13-192](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h118

**Table 13-192. INTPRIORITY\_114 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.191 INTPRIORITY\_115 Register (Offset = 11CCh) [Reset = 00000000h]

INTPRIORITY\_115 is shown in [Table 13-193](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h119

**Table 13-193. INTPRIORITY\_115 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.192 INTPRIORITY\_116 Register (Offset = 11D0h) [Reset = 00000000h]

INTPRIORITY\_116 is shown in [Table 13-194](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h120

**Table 13-194. INTPRIORITY\_116 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.193 INTPRIORITY\_117 Register (Offset = 11D4h) [Reset = 00000000h]

INTPRIORITY\_117 is shown in [Table 13-195](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h121

**Table 13-195. INTPRIORITY\_117 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.194 INTPRIORITY\_118 Register (Offset = 11D8h) [Reset = 00000000h]

INTPRIORITY\_118 is shown in [Table 13-196](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h122

**Table 13-196. INTPRIORITY\_118 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.195 INTPRIORITY\_119 Register (Offset = 11DCh) [Reset = 00000000h]

INTPRIORITY\_119 is shown in [Table 13-197](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h123

**Table 13-197. INTPRIORITY\_119 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.196 INTPRIORITY\_120 Register (Offset = 11E0h) [Reset = 00000000h]

INTPRIORITY\_120 is shown in [Table 13-198](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h124

**Table 13-198. INTPRIORITY\_120 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.197 INTPRIORITY\_121 Register (Offset = 11E4h) [Reset = 00000000h]

INTPRIORITY\_121 is shown in [Table 13-199](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h125

**Table 13-199. INTPRIORITY\_121 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.198 INTPRIORITY\_122 Register (Offset = 11E8h) [Reset = 00000000h]

INTPRIORITY\_122 is shown in [Table 13-200](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h126

**Table 13-200. INTPRIORITY\_122 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.199 INTPRIORITY\_123 Register (Offset = 11ECh) [Reset = 00000000h]

INTPRIORITY\_123 is shown in [Table 13-201](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h127

**Table 13-201. INTPRIORITY\_123 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.200 INTPRIORITY\_124 Register (Offset = 11F0h) [Reset = 00000000h]

INTPRIORITY\_124 is shown in [Table 13-202](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h128

**Table 13-202. INTPRIORITY\_124 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.201 INTPRIORITY\_125 Register (Offset = 11F4h) [Reset = 00000000h]

INTPRIORITY\_125 is shown in [Table 13-203](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h129

**Table 13-203. INTPRIORITY\_125 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.202 INTPRIORITY\_126 Register (Offset = 11F8h) [Reset = 00000000h]

INTPRIORITY\_126 is shown in [Table 13-204](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h130

**Table 13-204. INTPRIORITY\_126 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.203 INTPRIORITY\_127 Register (Offset = 11FCh) [Reset = 00000000h]

INTPRIORITY\_127 is shown in [Table 13-205](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h131

**Table 13-205. INTPRIORITY\_127 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.204 INTPRIORITY\_128 Register (Offset = 1200h) [Reset = 00000000h]

INTPRIORITY\_128 is shown in [Table 13-206](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h132

**Table 13-206. INTPRIORITY\_128 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.205 INTPRIORITY\_129 Register (Offset = 1204h) [Reset = 00000000h]

INTPRIORITY\_129 is shown in [Table 13-207](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h133

**Table 13-207. INTPRIORITY\_129 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.206 INTPRIORITY\_130 Register (Offset = 1208h) [Reset = 00000000h]

INTPRIORITY\_130 is shown in [Table 13-208](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h134

**Table 13-208. INTPRIORITY\_130 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.207 INTPRIORITY\_131 Register (Offset = 120Ch) [Reset = 00000000h]

INTPRIORITY\_131 is shown in [Table 13-209](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h135

**Table 13-209. INTPRIORITY\_131 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.208 INTPRIORITY\_132 Register (Offset = 1210h) [Reset = 00000000h]

INTPRIORITY\_132 is shown in [Table 13-210](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h136

**Table 13-210. INTPRIORITY\_132 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.209 INTPRIORITY\_133 Register (Offset = 1214h) [Reset = 00000000h]

INTPRIORITY\_133 is shown in [Table 13-211](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h137

**Table 13-211. INTPRIORITY\_133 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.210 INTPRIORITY\_134 Register (Offset = 1218h) [Reset = 00000000h]

INTPRIORITY\_134 is shown in [Table 13-212](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h138

**Table 13-212. INTPRIORITY\_134 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.211 INTPRIORITY\_135 Register (Offset = 121Ch) [Reset = 00000000h]

INTPRIORITY\_135 is shown in [Table 13-213](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h139

**Table 13-213. INTPRIORITY\_135 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.212 INTPRIORITY\_136 Register (Offset = 1220h) [Reset = 00000000h]

INTPRIORITY\_136 is shown in [Table 13-214](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h140

**Table 13-214. INTPRIORITY\_136 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.213 INTPRIORITY\_137 Register (Offset = 1224h) [Reset = 00000000h]

INTPRIORITY\_137 is shown in [Table 13-215](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h141

**Table 13-215. INTPRIORITY\_137 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.214 INTPRIORITY\_138 Register (Offset = 1228h) [Reset = 00000000h]

INTPRIORITY\_138 is shown in [Table 13-216](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h142

**Table 13-216. INTPRIORITY\_138 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.215 INTPRIORITY\_139 Register (Offset = 122Ch) [Reset = 00000000h]

INTPRIORITY\_139 is shown in [Table 13-217](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h143

**Table 13-217. INTPRIORITY\_139 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.216 INTPRIORITY\_140 Register (Offset = 1230h) [Reset = 00000000h]

INTPRIORITY\_140 is shown in [Table 13-218](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h144

**Table 13-218. INTPRIORITY\_140 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.217 INTPRIORITY\_141 Register (Offset = 1234h) [Reset = 00000000h]

INTPRIORITY\_141 is shown in [Table 13-219](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h145

**Table 13-219. INTPRIORITY\_141 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.218 INTPRIORITY\_142 Register (Offset = 1238h) [Reset = 00000000h]

INTPRIORITY\_142 is shown in [Table 13-220](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h146

**Table 13-220. INTPRIORITY\_142 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.219 INTPRIORITY\_143 Register (Offset = 123Ch) [Reset = 00000000h]

INTPRIORITY\_143 is shown in [Table 13-221](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h147

**Table 13-221. INTPRIORITY\_143 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.220 INTPRIORITY\_144 Register (Offset = 1240h) [Reset = 00000000h]

INTPRIORITY\_144 is shown in [Table 13-222](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h148

**Table 13-222. INTPRIORITY\_144 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.221 INTPRIORITY\_145 Register (Offset = 1244h) [Reset = 00000000h]

INTPRIORITY\_145 is shown in [Table 13-223](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h149

**Table 13-223. INTPRIORITY\_145 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.222 INTPRIORITY\_146 Register (Offset = 1248h) [Reset = 00000000h]

INTPRIORITY\_146 is shown in [Table 13-224](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h150

**Table 13-224. INTPRIORITY\_146 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.223 INTPRIORITY\_147 Register (Offset = 124Ch) [Reset = 00000000h]

INTPRIORITY\_147 is shown in [Table 13-225](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h151

**Table 13-225. INTPRIORITY\_147 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.224 INTPRIORITY\_148 Register (Offset = 1250h) [Reset = 00000000h]

INTPRIORITY\_148 is shown in [Table 13-226](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h152

**Table 13-226. INTPRIORITY\_148 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.225 INTPRIORITY\_149 Register (Offset = 1254h) [Reset = 00000000h]

INTPRIORITY\_149 is shown in [Table 13-227](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h153

**Table 13-227. INTPRIORITY\_149 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.226 INTPRIORITY\_150 Register (Offset = 1258h) [Reset = 00000000h]

INTPRIORITY\_150 is shown in [Table 13-228](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h154

**Table 13-228. INTPRIORITY\_150 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.227 INTPRIORITY\_151 Register (Offset = 125Ch) [Reset = 00000000h]

INTPRIORITY\_151 is shown in [Table 13-229](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h155

**Table 13-229. INTPRIORITY\_151 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.228 INTPRIORITY\_152 Register (Offset = 1260h) [Reset = 00000000h]

INTPRIORITY\_152 is shown in [Table 13-230](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h156

**Table 13-230. INTPRIORITY\_152 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.229 INTPRIORITY\_153 Register (Offset = 1264h) [Reset = 00000000h]

INTPRIORITY\_153 is shown in [Table 13-231](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h157

**Table 13-231. INTPRIORITY\_153 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.230 INTPRIORITY\_154 Register (Offset = 1268h) [Reset = 00000000h]

INTPRIORITY\_154 is shown in [Table 13-232](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h158

**Table 13-232. INTPRIORITY\_154 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.231 INTPRIORITY\_155 Register (Offset = 126Ch) [Reset = 00000000h]

INTPRIORITY\_155 is shown in [Table 13-233](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h159

**Table 13-233. INTPRIORITY\_155 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.232 INTPRIORITY\_156 Register (Offset = 1270h) [Reset = 00000000h]

INTPRIORITY\_156 is shown in [Table 13-234](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h160

**Table 13-234. INTPRIORITY\_156 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.233 INTPRIORITY\_157 Register (Offset = 1274h) [Reset = 00000000h]

INTPRIORITY\_157 is shown in [Table 13-235](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h161

**Table 13-235. INTPRIORITY\_157 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.234 INTPRIORITY\_158 Register (Offset = 1278h) [Reset = 00000000h]

INTPRIORITY\_158 is shown in [Table 13-236](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h162

**Table 13-236. INTPRIORITY\_158 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.235 INTPRIORITY\_159 Register (Offset = 127Ch) [Reset = 00000000h]

INTPRIORITY\_159 is shown in [Table 13-237](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h163

**Table 13-237. INTPRIORITY\_159 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.236 INTPRIORITY\_160 Register (Offset = 1280h) [Reset = 00000000h]

INTPRIORITY\_160 is shown in [Table 13-238](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h164

**Table 13-238. INTPRIORITY\_160 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.237 INTPRIORITY\_161 Register (Offset = 1284h) [Reset = 00000000h]

INTPRIORITY\_161 is shown in [Table 13-239](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h165

**Table 13-239. INTPRIORITY\_161 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.238 INTPRIORITY\_162 Register (Offset = 1288h) [Reset = 00000000h]

INTPRIORITY\_162 is shown in [Table 13-240](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h166

**Table 13-240. INTPRIORITY\_162 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.239 INTPRIORITY\_163 Register (Offset = 128Ch) [Reset = 00000000h]

INTPRIORITY\_163 is shown in [Table 13-241](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h167

**Table 13-241. INTPRIORITY\_163 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.240 INTPRIORITY\_164 Register (Offset = 1290h) [Reset = 00000000h]

INTPRIORITY\_164 is shown in [Table 13-242](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h168

**Table 13-242. INTPRIORITY\_164 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.241 INTPRIORITY\_165 Register (Offset = 1294h) [Reset = 00000000h]

INTPRIORITY\_165 is shown in [Table 13-243](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h169

**Table 13-243. INTPRIORITY\_165 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.242 INTPRIORITY\_166 Register (Offset = 1298h) [Reset = 00000000h]

INTPRIORITY\_166 is shown in [Table 13-244](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h170

**Table 13-244. INTPRIORITY\_166 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.243 INTPRIORITY\_167 Register (Offset = 129Ch) [Reset = 00000000h]

INTPRIORITY\_167 is shown in [Table 13-245](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h171

**Table 13-245. INTPRIORITY\_167 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.244 INTPRIORITY\_168 Register (Offset = 12A0h) [Reset = 00000000h]

INTPRIORITY\_168 is shown in [Table 13-246](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h172

**Table 13-246. INTPRIORITY\_168 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.245 INTPRIORITY\_169 Register (Offset = 12A4h) [Reset = 00000000h]

INTPRIORITY\_169 is shown in [Table 13-247](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h173

**Table 13-247. INTPRIORITY\_169 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.246 INTPRIORITY\_170 Register (Offset = 12A8h) [Reset = 00000000h]

INTPRIORITY\_170 is shown in [Table 13-248](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h174

**Table 13-248. INTPRIORITY\_170 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.247 INTPRIORITY\_171 Register (Offset = 12ACh) [Reset = 00000000h]

INTPRIORITY\_171 is shown in [Table 13-249](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h175

**Table 13-249. INTPRIORITY\_171 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.248 INTPRIORITY\_172 Register (Offset = 12B0h) [Reset = 00000000h]

INTPRIORITY\_172 is shown in [Table 13-250](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h176

**Table 13-250. INTPRIORITY\_172 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.249 INTPRIORITY\_173 Register (Offset = 12B4h) [Reset = 00000000h]

INTPRIORITY\_173 is shown in [Table 13-251](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h177

**Table 13-251. INTPRIORITY\_173 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.250 INTPRIORITY\_174 Register (Offset = 12B8h) [Reset = 00000000h]

INTPRIORITY\_174 is shown in [Table 13-252](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h178

**Table 13-252. INTPRIORITY\_174 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.251 INTPRIORITY\_175 Register (Offset = 12BCh) [Reset = 00000000h]

INTPRIORITY\_175 is shown in [Table 13-253](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h179

**Table 13-253. INTPRIORITY\_175 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.252 INTPRIORITY\_176 Register (Offset = 12C0h) [Reset = 00000000h]

INTPRIORITY\_176 is shown in [Table 13-254](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h180

**Table 13-254. INTPRIORITY\_176 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.253 INTPRIORITY\_177 Register (Offset = 12C4h) [Reset = 00000000h]

INTPRIORITY\_177 is shown in [Table 13-255](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h181

**Table 13-255. INTPRIORITY\_177 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.254 INTPRIORITY\_178 Register (Offset = 12C8h) [Reset = 00000000h]

INTPRIORITY\_178 is shown in [Table 13-256](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h182

**Table 13-256. INTPRIORITY\_178 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.255 INTPRIORITY\_179 Register (Offset = 12CCh) [Reset = 00000000h]

INTPRIORITY\_179 is shown in [Table 13-257](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h183

**Table 13-257. INTPRIORITY\_179 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.256 INTPRIORITY\_180 Register (Offset = 12D0h) [Reset = 00000000h]

INTPRIORITY\_180 is shown in [Table 13-258](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h184

**Table 13-258. INTPRIORITY\_180 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.257 INTPRIORITY\_181 Register (Offset = 12D4h) [Reset = 00000000h]

INTPRIORITY\_181 is shown in [Table 13-259](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h185

**Table 13-259. INTPRIORITY\_181 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.258 INTPRIORITY\_182 Register (Offset = 12D8h) [Reset = 00000000h]

INTPRIORITY\_182 is shown in [Table 13-260](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h186

**Table 13-260. INTPRIORITY\_182 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.259 INTPRIORITY\_183 Register (Offset = 12DCh) [Reset = 00000000h]

INTPRIORITY\_183 is shown in [Table 13-261](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h187

**Table 13-261. INTPRIORITY\_183 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.260 INTPRIORITY\_184 Register (Offset = 12E0h) [Reset = 00000000h]

INTPRIORITY\_184 is shown in [Table 13-262](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h188

**Table 13-262. INTPRIORITY\_184 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.261 INTPRIORITY\_185 Register (Offset = 12E4h) [Reset = 00000000h]

INTPRIORITY\_185 is shown in [Table 13-263](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h189

**Table 13-263. INTPRIORITY\_185 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.262 INTPRIORITY\_186 Register (Offset = 12E8h) [Reset = 00000000h]

INTPRIORITY\_186 is shown in [Table 13-264](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h190

**Table 13-264. INTPRIORITY\_186 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.263 INTPRIORITY\_187 Register (Offset = 12ECh) [Reset = 00000000h]

INTPRIORITY\_187 is shown in [Table 13-265](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h191

**Table 13-265. INTPRIORITY\_187 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.264 INTPRIORITY\_188 Register (Offset = 12F0h) [Reset = 00000000h]

INTPRIORITY\_188 is shown in [Table 13-266](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h192

**Table 13-266. INTPRIORITY\_188 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.265 INTPRIORITY\_189 Register (Offset = 12F4h) [Reset = 00000000h]

INTPRIORITY\_189 is shown in [Table 13-267](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h193

**Table 13-267. INTPRIORITY\_189 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.266 INTPRIORITY\_190 Register (Offset = 12F8h) [Reset = 00000000h]

INTPRIORITY\_190 is shown in [Table 13-268](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h194

**Table 13-268. INTPRIORITY\_190 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.267 INTPRIORITY\_191 Register (Offset = 12FCh) [Reset = 00000000h]

INTPRIORITY\_191 is shown in [Table 13-269](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h195

**Table 13-269. INTPRIORITY\_191 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.268 INTPRIORITY\_192 Register (Offset = 1300h) [Reset = 00000000h]

INTPRIORITY\_192 is shown in [Table 13-270](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h196

**Table 13-270. INTPRIORITY\_192 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.269 INTPRIORITY\_193 Register (Offset = 1304h) [Reset = 00000000h]

INTPRIORITY\_193 is shown in [Table 13-271](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h197

**Table 13-271. INTPRIORITY\_193 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.270 INTPRIORITY\_194 Register (Offset = 1308h) [Reset = 00000000h]

INTPRIORITY\_194 is shown in [Table 13-272](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h198

**Table 13-272. INTPRIORITY\_194 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.271 INTPRIORITY\_195 Register (Offset = 130Ch) [Reset = 00000000h]

INTPRIORITY\_195 is shown in [Table 13-273](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h199

**Table 13-273. INTPRIORITY\_195 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.272 INTPRIORITY\_196 Register (Offset = 1310h) [Reset = 00000000h]

INTPRIORITY\_196 is shown in [Table 13-274](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h200

**Table 13-274. INTPRIORITY\_196 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.273 INTPRIORITY\_197 Register (Offset = 1314h) [Reset = 00000000h]

INTPRIORITY\_197 is shown in [Table 13-275](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h201

**Table 13-275. INTPRIORITY\_197 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.274 INTPRIORITY\_198 Register (Offset = 1318h) [Reset = 00000000h]

INTPRIORITY\_198 is shown in [Table 13-276](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h202

**Table 13-276. INTPRIORITY\_198 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.275 INTPRIORITY\_199 Register (Offset = 131Ch) [Reset = 00000000h]

INTPRIORITY\_199 is shown in [Table 13-277](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h203

**Table 13-277. INTPRIORITY\_199 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.276 INTPRIORITY\_200 Register (Offset = 1320h) [Reset = 00000000h]

INTPRIORITY\_200 is shown in [Table 13-278](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h204

**Table 13-278. INTPRIORITY\_200 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.277 INTPRIORITY\_201 Register (Offset = 1324h) [Reset = 00000000h]

INTPRIORITY\_201 is shown in [Table 13-279](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h205

**Table 13-279. INTPRIORITY\_201 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.278 INTPRIORITY\_202 Register (Offset = 1328h) [Reset = 00000000h]

INTPRIORITY\_202 is shown in [Table 13-280](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h206

**Table 13-280. INTPRIORITY\_202 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.279 INTPRIORITY\_203 Register (Offset = 132Ch) [Reset = 00000000h]

INTPRIORITY\_203 is shown in [Table 13-281](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h207

**Table 13-281. INTPRIORITY\_203 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.280 INTPRIORITY\_204 Register (Offset = 1330h) [Reset = 00000000h]

INTPRIORITY\_204 is shown in [Table 13-282](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h208

**Table 13-282. INTPRIORITY\_204 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.281 INTPRIORITY\_205 Register (Offset = 1334h) [Reset = 00000000h]

INTPRIORITY\_205 is shown in [Table 13-283](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h209

**Table 13-283. INTPRIORITY\_205 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.282 INTPRIORITY\_206 Register (Offset = 1338h) [Reset = 00000000h]

INTPRIORITY\_206 is shown in [Table 13-284](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h210

**Table 13-284. INTPRIORITY\_206 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.283 INTPRIORITY\_207 Register (Offset = 133Ch) [Reset = 00000000h]

INTPRIORITY\_207 is shown in [Table 13-285](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h211

**Table 13-285. INTPRIORITY\_207 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.284 INTPRIORITY\_208 Register (Offset = 1340h) [Reset = 00000000h]

INTPRIORITY\_208 is shown in [Table 13-286](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h212

**Table 13-286. INTPRIORITY\_208 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.285 INTPRIORITY\_209 Register (Offset = 1344h) [Reset = 00000000h]

INTPRIORITY\_209 is shown in [Table 13-287](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h213

**Table 13-287. INTPRIORITY\_209 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.286 INTPRIORITY\_210 Register (Offset = 1348h) [Reset = 00000000h]

INTPRIORITY\_210 is shown in [Table 13-288](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h214

**Table 13-288. INTPRIORITY\_210 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.287 INTPRIORITY\_211 Register (Offset = 134Ch) [Reset = 00000000h]

INTPRIORITY\_211 is shown in [Table 13-289](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h215

**Table 13-289. INTPRIORITY\_211 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.288 INTPRIORITY\_212 Register (Offset = 1350h) [Reset = 00000000h]

INTPRIORITY\_212 is shown in [Table 13-290](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h216

**Table 13-290. INTPRIORITY\_212 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.289 INTPRIORITY\_213 Register (Offset = 1354h) [Reset = 00000000h]

INTPRIORITY\_213 is shown in [Table 13-291](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h217

**Table 13-291. INTPRIORITY\_213 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.290 INTPRIORITY\_214 Register (Offset = 1358h) [Reset = 00000000h]

INTPRIORITY\_214 is shown in [Table 13-292](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h218

**Table 13-292. INTPRIORITY\_214 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.291 INTPRIORITY\_215 Register (Offset = 135Ch) [Reset = 00000000h]

INTPRIORITY\_215 is shown in [Table 13-293](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h219

**Table 13-293. INTPRIORITY\_215 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.292 INTPRIORITY\_216 Register (Offset = 1360h) [Reset = 00000000h]

INTPRIORITY\_216 is shown in [Table 13-294](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h220

**Table 13-294. INTPRIORITY\_216 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.293 INTPRIORITY\_217 Register (Offset = 1364h) [Reset = 00000000h]

INTPRIORITY\_217 is shown in [Table 13-295](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h221

**Table 13-295. INTPRIORITY\_217 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.294 INTPRIORITY\_218 Register (Offset = 1368h) [Reset = 00000000h]

INTPRIORITY\_218 is shown in [Table 13-296](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h222

**Table 13-296. INTPRIORITY\_218 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.295 INTPRIORITY\_219 Register (Offset = 136Ch) [Reset = 00000000h]

INTPRIORITY\_219 is shown in [Table 13-297](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h223

**Table 13-297. INTPRIORITY\_219 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.296 INTPRIORITY\_220 Register (Offset = 1370h) [Reset = 00000000h]

INTPRIORITY\_220 is shown in [Table 13-298](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h224

**Table 13-298. INTPRIORITY\_220 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.297 INTPRIORITY\_221 Register (Offset = 1374h) [Reset = 00000000h]

INTPRIORITY\_221 is shown in [Table 13-299](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h225

**Table 13-299. INTPRIORITY\_221 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.298 INTPRIORITY\_222 Register (Offset = 1378h) [Reset = 00000000h]

INTPRIORITY\_222 is shown in [Table 13-300](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h226

**Table 13-300. INTPRIORITY\_222 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.299 INTPRIORITY\_223 Register (Offset = 137Ch) [Reset = 00000000h]

INTPRIORITY\_223 is shown in [Table 13-301](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h227

**Table 13-301. INTPRIORITY\_223 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.300 INTPRIORITY\_224 Register (Offset = 1380h) [Reset = 00000000h]

INTPRIORITY\_224 is shown in [Table 13-302](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h228

**Table 13-302. INTPRIORITY\_224 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.301 INTPRIORITY\_225 Register (Offset = 1384h) [Reset = 00000000h]

INTPRIORITY\_225 is shown in [Table 13-303](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h229

**Table 13-303. INTPRIORITY\_225 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.302 INTPRIORITY\_226 Register (Offset = 1388h) [Reset = 00000000h]

INTPRIORITY\_226 is shown in [Table 13-304](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h230

**Table 13-304. INTPRIORITY\_226 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.303 INTPRIORITY\_227 Register (Offset = 138Ch) [Reset = 00000000h]

INTPRIORITY\_227 is shown in [Table 13-305](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h231

**Table 13-305. INTPRIORITY\_227 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.304 INTPRIORITY\_228 Register (Offset = 1390h) [Reset = 00000000h]

INTPRIORITY\_228 is shown in [Table 13-306](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h232

**Table 13-306. INTPRIORITY\_228 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.305 INTPRIORITY\_229 Register (Offset = 1394h) [Reset = 00000000h]

INTPRIORITY\_229 is shown in [Table 13-307](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h233

**Table 13-307. INTPRIORITY\_229 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.306 INTPRIORITY\_230 Register (Offset = 1398h) [Reset = 00000000h]

INTPRIORITY\_230 is shown in [Table 13-308](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h234

**Table 13-308. INTPRIORITY\_230 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.307 INTPRIORITY\_231 Register (Offset = 139Ch) [Reset = 00000000h]

INTPRIORITY\_231 is shown in [Table 13-309](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h235

**Table 13-309. INTPRIORITY\_231 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.308 INTPRIORITY\_232 Register (Offset = 13A0h) [Reset = 00000000h]

INTPRIORITY\_232 is shown in [Table 13-310](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h236

**Table 13-310. INTPRIORITY\_232 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.309 INTPRIORITY\_233 Register (Offset = 13A4h) [Reset = 00000000h]

INTPRIORITY\_233 is shown in [Table 13-311](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h237

**Table 13-311. INTPRIORITY\_233 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.310 INTPRIORITY\_234 Register (Offset = 13A8h) [Reset = 00000000h]

INTPRIORITY\_234 is shown in [Table 13-312](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h238

**Table 13-312. INTPRIORITY\_234 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.311 INTPRIORITY\_235 Register (Offset = 13ACh) [Reset = 00000000h]

INTPRIORITY\_235 is shown in [Table 13-313](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h239

**Table 13-313. INTPRIORITY\_235 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.312 INTPRIORITY\_236 Register (Offset = 13B0h) [Reset = 00000000h]

INTPRIORITY\_236 is shown in [Table 13-314](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h240

**Table 13-314. INTPRIORITY\_236 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.313 INTPRIORITY\_237 Register (Offset = 13B4h) [Reset = 00000000h]

INTPRIORITY\_237 is shown in [Table 13-315](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h241

**Table 13-315. INTPRIORITY\_237 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.314 INTPRIORITY\_238 Register (Offset = 13B8h) [Reset = 00000000h]

INTPRIORITY\_238 is shown in [Table 13-316](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h242

**Table 13-316. INTPRIORITY\_238 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.315 INTPRIORITY\_239 Register (Offset = 13BCh) [Reset = 00000000h]

INTPRIORITY\_239 is shown in [Table 13-317](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h243

**Table 13-317. INTPRIORITY\_239 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.316 INTPRIORITY\_240 Register (Offset = 13C0h) [Reset = 00000000h]

INTPRIORITY\_240 is shown in [Table 13-318](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h244

**Table 13-318. INTPRIORITY\_240 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.317 INTPRIORITY\_241 Register (Offset = 13C4h) [Reset = 00000000h]

INTPRIORITY\_241 is shown in [Table 13-319](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h245

**Table 13-319. INTPRIORITY\_241 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.318 INTPRIORITY\_242 Register (Offset = 13C8h) [Reset = 00000000h]

INTPRIORITY\_242 is shown in [Table 13-320](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h246

**Table 13-320. INTPRIORITY\_242 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.319 INTPRIORITY\_243 Register (Offset = 13CCh) [Reset = 00000000h]

INTPRIORITY\_243 is shown in [Table 13-321](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h247

**Table 13-321. INTPRIORITY\_243 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.320 INTPRIORITY\_244 Register (Offset = 13D0h) [Reset = 00000000h]

INTPRIORITY\_244 is shown in [Table 13-322](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h248

**Table 13-322. INTPRIORITY\_244 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.321 INTPRIORITY\_245 Register (Offset = 13D4h) [Reset = 00000000h]

INTPRIORITY\_245 is shown in [Table 13-323](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h249

**Table 13-323. INTPRIORITY\_245 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.322 INTPRIORITY\_246 Register (Offset = 13D8h) [Reset = 00000000h]

INTPRIORITY\_246 is shown in [Table 13-324](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h250

**Table 13-324. INTPRIORITY\_246 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.323 INTPRIORITY\_247 Register (Offset = 13DCh) [Reset = 00000000h]

INTPRIORITY\_247 is shown in [Table 13-325](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h251

**Table 13-325. INTPRIORITY\_247 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.324 INTPRIORITY\_248 Register (Offset = 13E0h) [Reset = 00000000h]

INTPRIORITY\_248 is shown in [Table 13-326](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h252

**Table 13-326. INTPRIORITY\_248 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.325 INTPRIORITY\_249 Register (Offset = 13E4h) [Reset = 00000000h]

INTPRIORITY\_249 is shown in [Table 13-327](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h253

**Table 13-327. INTPRIORITY\_249 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)



### 13.10.326 INTPRIORITY\_250 Register (Offset = 13E8h) [Reset = 00000000h]

INTPRIORITY\_250 is shown in [Table 13-328](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h254

**Table 13-328. INTPRIORITY\_250 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.327 INTPRIORITY\_251 Register (Offset = 13ECh) [Reset = 00000000h]

INTPRIORITY\_251 is shown in [Table 13-329](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h255

**Table 13-329. INTPRIORITY\_251 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.328 INTPRIORITY\_252 Register (Offset = 13F0h) [Reset = 00000000h]

INTPRIORITY\_252 is shown in [Table 13-330](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h256

**Table 13-330. INTPRIORITY\_252 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.329 INTPRIORITY\_253 Register (Offset = 13F4h) [Reset = 00000000h]

INTPRIORITY\_253 is shown in [Table 13-331](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h257

**Table 13-331. INTPRIORITY\_253 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.330 INTPRIORITY\_254 Register (Offset = 13F8h) [Reset = 00000000h]

INTPRIORITY\_254 is shown in [Table 13-332](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h258

**Table 13-332. INTPRIORITY\_254 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.331 INTPRIORITY\_255 Register (Offset = 13FCh) [Reset = 00000000h]

INTPRIORITY\_255 is shown in [Table 13-333](#).

Return to the [Summary Table](#).

Interrupt Q Priority Register (Q is 0 to 255 , Q= M+1 x 32) h1000 + Q x h259

**Table 13-333. INTPRIORITY\_255 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RES19	R	0h	RESERVE FIELD
3-0	PRI	R/W	Fh	This is the priority for interrupt Q. If two interrupts have the same priority, then whichever interrupt has the lower number Q wins arbitration 0 Highest Priority 15 Lowest Priority (Default)

### 13.10.332 INTVECTOR Register (Offset = 2000h) [Reset = 00000000h]

INTVECTOR is shown in [Table 13-334](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h4

**Table 13-334. INTVECTOR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.333 INTVECTOR\_1 Register (Offset = 2004h) [Reset = 00000000h]

INTVECTOR\_1 is shown in [Table 13-335](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h5

**Table 13-335. INTVECTOR\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.334 INTVECTOR\_2 Register (Offset = 2008h) [Reset = 00000000h]

INTVECTOR\_2 is shown in [Table 13-336](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h6

**Table 13-336. INTVECTOR\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.335 INTVECTOR\_3 Register (Offset = 200Ch) [Reset = 00000000h]

INTVECTOR\_3 is shown in [Table 13-337](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h7

**Table 13-337. INTVECTOR\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.336 INTVECTOR\_4 Register (Offset = 2010h) [Reset = 00000000h]

INTVECTOR\_4 is shown in [Table 13-338](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h8

**Table 13-338. INTVECTOR\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.337 INTVECTOR\_5 Register (Offset = 2014h) [Reset = 00000000h]

INTVECTOR\_5 is shown in [Table 13-339](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h9

**Table 13-339. INTVECTOR\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.338 INTVECTOR\_6 Register (Offset = 2018h) [Reset = 00000000h]

INTVECTOR\_6 is shown in [Table 13-340](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h10

**Table 13-340. INTVECTOR\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.339 INTVECTOR\_7 Register (Offset = 201Ch) [Reset = 00000000h]

INTVECTOR\_7 is shown in [Table 13-341](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h11

**Table 13-341. INTVECTOR\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.340 INTVECTOR\_8 Register (Offset = 2020h) [Reset = 00000000h]

INTVECTOR\_8 is shown in [Table 13-342](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h12

**Table 13-342. INTVECTOR\_8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.341 INTVECTOR\_9 Register (Offset = 2024h) [Reset = 00000000h]

INTVECTOR\_9 is shown in [Table 13-343](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h13

**Table 13-343. INTVECTOR\_9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.342 INTVECTOR\_10 Register (Offset = 2028h) [Reset = 00000000h]

INTVECTOR\_10 is shown in [Table 13-344](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h14

**Table 13-344. INTVECTOR\_10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.343 INTVECTOR\_11 Register (Offset = 202Ch) [Reset = 00000000h]

INTVECTOR\_11 is shown in [Table 13-345](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h15

**Table 13-345. INTVECTOR\_11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.344 INTVECTOR\_12 Register (Offset = 2030h) [Reset = 00000000h]

INTVECTOR\_12 is shown in [Table 13-346](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h16

**Table 13-346. INTVECTOR\_12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.345 INTVECTOR\_13 Register (Offset = 2034h) [Reset = 00000000h]

INTVECTOR\_13 is shown in [Table 13-347](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h17

**Table 13-347. INTVECTOR\_13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.346 INTVECTOR\_14 Register (Offset = 2038h) [Reset = 00000000h]

INTVECTOR\_14 is shown in [Table 13-348](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h18

**Table 13-348. INTVECTOR\_14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.347 INTVECTOR\_15 Register (Offset = 203Ch) [Reset = 00000000h]

INTVECTOR\_15 is shown in [Table 13-349](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h19

**Table 13-349. INTVECTOR\_15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.348 INTVECTOR\_16 Register (Offset = 2040h) [Reset = 00000000h]

INTVECTOR\_16 is shown in [Table 13-350](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h20

**Table 13-350. INTVECTOR\_16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.349 INTVECTOR\_17 Register (Offset = 2044h) [Reset = 00000000h]

INTVECTOR\_17 is shown in [Table 13-351](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h21

**Table 13-351. INTVECTOR\_17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.350 INTVECTOR\_18 Register (Offset = 2048h) [Reset = 00000000h]

INTVECTOR\_18 is shown in [Table 13-352](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h22

**Table 13-352. INTVECTOR\_18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.351 INTVECTOR\_19 Register (Offset = 204Ch) [Reset = 00000000h]

INTVECTOR\_19 is shown in [Table 13-353](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h23

**Table 13-353. INTVECTOR\_19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.352 INTVECTOR\_20 Register (Offset = 2050h) [Reset = 00000000h]

INTVECTOR\_20 is shown in [Table 13-354](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h24

**Table 13-354. INTVECTOR\_20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.353 INTVECTOR\_21 Register (Offset = 2054h) [Reset = 00000000h]

INTVECTOR\_21 is shown in [Table 13-355](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h25

**Table 13-355. INTVECTOR\_21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.354 INTVECTOR\_22 Register (Offset = 2058h) [Reset = 00000000h]

INTVECTOR\_22 is shown in [Table 13-356](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h26

**Table 13-356. INTVECTOR\_22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.355 INTVECTOR\_23 Register (Offset = 205Ch) [Reset = 00000000h]

INTVECTOR\_23 is shown in [Table 13-357](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h27

**Table 13-357. INTVECTOR\_23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.356 INTVECTOR\_24 Register (Offset = 2060h) [Reset = 00000000h]

INTVECTOR\_24 is shown in [Table 13-358](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h28

**Table 13-358. INTVECTOR\_24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.357 INTVECTOR\_25 Register (Offset = 2064h) [Reset = 00000000h]

INTVECTOR\_25 is shown in [Table 13-359](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h29

**Table 13-359. INTVECTOR\_25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.358 INTVECTOR\_26 Register (Offset = 2068h) [Reset = 00000000h]

INTVECTOR\_26 is shown in [Table 13-360](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h30

**Table 13-360. INTVECTOR\_26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.359 INTVECTOR\_27 Register (Offset = 206Ch) [Reset = 00000000h]

INTVECTOR\_27 is shown in [Table 13-361](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h31

**Table 13-361. INTVECTOR\_27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.360 INTVECTOR\_28 Register (Offset = 2070h) [Reset = 00000000h]

INTVECTOR\_28 is shown in [Table 13-362](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h32

**Table 13-362. INTVECTOR\_28 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.361 INTVECTOR\_29 Register (Offset = 2074h) [Reset = 00000000h]

INTVECTOR\_29 is shown in [Table 13-363](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h33

**Table 13-363. INTVECTOR\_29 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.362 INTVECTOR\_30 Register (Offset = 2078h) [Reset = 00000000h]

INTVECTOR\_30 is shown in [Table 13-364](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h34

**Table 13-364. INTVECTOR\_30 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.363 INTVECTOR\_31 Register (Offset = 207Ch) [Reset = 00000000h]

INTVECTOR\_31 is shown in [Table 13-365](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h35

**Table 13-365. INTVECTOR\_31 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.364 INTVECTOR\_32 Register (Offset = 2080h) [Reset = 00000000h]

INTVECTOR\_32 is shown in [Table 13-366](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h36

**Table 13-366. INTVECTOR\_32 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.365 INTVECTOR\_33 Register (Offset = 2084h) [Reset = 00000000h]

INTVECTOR\_33 is shown in [Table 13-367](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h37

**Table 13-367. INTVECTOR\_33 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.366 INTVECTOR\_34 Register (Offset = 2088h) [Reset = 00000000h]

INTVECTOR\_34 is shown in [Table 13-368](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h38

**Table 13-368. INTVECTOR\_34 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.367 INTVECTOR\_35 Register (Offset = 208Ch) [Reset = 00000000h]

INTVECTOR\_35 is shown in [Table 13-369](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h39

**Table 13-369. INTVECTOR\_35 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.368 INTVECTOR\_36 Register (Offset = 2090h) [Reset = 00000000h]

INTVECTOR\_36 is shown in [Table 13-370](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h40

**Table 13-370. INTVECTOR\_36 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.369 INTVECTOR\_37 Register (Offset = 2094h) [Reset = 00000000h]

INTVECTOR\_37 is shown in [Table 13-371](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h41

**Table 13-371. INTVECTOR\_37 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.370 INTVECTOR\_38 Register (Offset = 2098h) [Reset = 00000000h]

INTVECTOR\_38 is shown in [Table 13-372](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h42

**Table 13-372. INTVECTOR\_38 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.371 INTVECTOR\_39 Register (Offset = 209Ch) [Reset = 00000000h]

INTVECTOR\_39 is shown in [Table 13-373](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h43

**Table 13-373. INTVECTOR\_39 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.372 INTVECTOR\_40 Register (Offset = 20A0h) [Reset = 00000000h]

INTVECTOR\_40 is shown in [Table 13-374](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h44

**Table 13-374. INTVECTOR\_40 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.373 INTVECTOR\_41 Register (Offset = 20A4h) [Reset = 00000000h]

INTVECTOR\_41 is shown in [Table 13-375](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h45

**Table 13-375. INTVECTOR\_41 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.374 INTVECTOR\_42 Register (Offset = 20A8h) [Reset = 00000000h]

INTVECTOR\_42 is shown in [Table 13-376](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h46

**Table 13-376. INTVECTOR\_42 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.375 INTVECTOR\_43 Register (Offset = 20ACh) [Reset = 00000000h]

INTVECTOR\_43 is shown in [Table 13-377](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h47

**Table 13-377. INTVECTOR\_43 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.376 INTVECTOR\_44 Register (Offset = 20B0h) [Reset = 00000000h]

INTVECTOR\_44 is shown in [Table 13-378](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h48

**Table 13-378. INTVECTOR\_44 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.377 INTVECTOR\_45 Register (Offset = 20B4h) [Reset = 00000000h]

INTVECTOR\_45 is shown in [Table 13-379](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h49

**Table 13-379. INTVECTOR\_45 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.378 INTVECTOR\_46 Register (Offset = 20B8h) [Reset = 00000000h]

INTVECTOR\_46 is shown in [Table 13-380](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h50

**Table 13-380. INTVECTOR\_46 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.379 INTVECTOR\_47 Register (Offset = 20BCh) [Reset = 00000000h]

INTVECTOR\_47 is shown in [Table 13-381](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h51

**Table 13-381. INTVECTOR\_47 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.380 INTVECTOR\_48 Register (Offset = 20C0h) [Reset = 00000000h]

INTVECTOR\_48 is shown in [Table 13-382](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h52

**Table 13-382. INTVECTOR\_48 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.381 INTVECTOR\_49 Register (Offset = 20C4h) [Reset = 00000000h]

INTVECTOR\_49 is shown in [Table 13-383](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h53

**Table 13-383. INTVECTOR\_49 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.382 INTVECTOR\_50 Register (Offset = 20C8h) [Reset = 00000000h]

INTVECTOR\_50 is shown in [Table 13-384](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h54

**Table 13-384. INTVECTOR\_50 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.383 INTVECTOR\_51 Register (Offset = 20CCh) [Reset = 00000000h]

INTVECTOR\_51 is shown in [Table 13-385](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h55

**Table 13-385. INTVECTOR\_51 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.384 INTVECTOR\_52 Register (Offset = 20D0h) [Reset = 00000000h]

INTVECTOR\_52 is shown in [Table 13-386](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h56

**Table 13-386. INTVECTOR\_52 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.385 INTVECTOR\_53 Register (Offset = 20D4h) [Reset = 00000000h]

INTVECTOR\_53 is shown in [Table 13-387](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h57

**Table 13-387. INTVECTOR\_53 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.386 INTVECTOR\_54 Register (Offset = 20D8h) [Reset = 00000000h]

INTVECTOR\_54 is shown in [Table 13-388](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h58

**Table 13-388. INTVECTOR\_54 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.387 INTVECTOR\_55 Register (Offset = 20DCh) [Reset = 00000000h]

INTVECTOR\_55 is shown in [Table 13-389](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h59

**Table 13-389. INTVECTOR\_55 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.388 INTVECTOR\_56 Register (Offset = 20E0h) [Reset = 00000000h]

INTVECTOR\_56 is shown in [Table 13-390](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h60

**Table 13-390. INTVECTOR\_56 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.389 INTVECTOR\_57 Register (Offset = 20E4h) [Reset = 00000000h]

INTVECTOR\_57 is shown in [Table 13-391](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h61

**Table 13-391. INTVECTOR\_57 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.390 INTVECTOR\_58 Register (Offset = 20E8h) [Reset = 00000000h]

INTVECTOR\_58 is shown in [Table 13-392](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h62

**Table 13-392. INTVECTOR\_58 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.391 INTVECTOR\_59 Register (Offset = 20ECh) [Reset = 00000000h]

INTVECTOR\_59 is shown in [Table 13-393](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h63

**Table 13-393. INTVECTOR\_59 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.392 INTVECTOR\_60 Register (Offset = 20F0h) [Reset = 00000000h]

INTVECTOR\_60 is shown in [Table 13-394](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h64

**Table 13-394. INTVECTOR\_60 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.393 INTVECTOR\_61 Register (Offset = 20F4h) [Reset = 00000000h]

INTVECTOR\_61 is shown in [Table 13-395](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h65

**Table 13-395. INTVECTOR\_61 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.394 INTVECTOR\_62 Register (Offset = 20F8h) [Reset = 00000000h]

INTVECTOR\_62 is shown in [Table 13-396](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h66

**Table 13-396. INTVECTOR\_62 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.395 INTVECTOR\_63 Register (Offset = 20FCh) [Reset = 00000000h]

INTVECTOR\_63 is shown in [Table 13-397](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h67

**Table 13-397. INTVECTOR\_63 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.396 INTVECTOR\_64 Register (Offset = 2100h) [Reset = 00000000h]

INTVECTOR\_64 is shown in [Table 13-398](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h68

**Table 13-398. INTVECTOR\_64 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.397 INTVECTOR\_65 Register (Offset = 2104h) [Reset = 00000000h]

INTVECTOR\_65 is shown in [Table 13-399](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h69

**Table 13-399. INTVECTOR\_65 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.398 INTVECTOR\_66 Register (Offset = 2108h) [Reset = 00000000h]

INTVECTOR\_66 is shown in [Table 13-400](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h70

**Table 13-400. INTVECTOR\_66 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.399 INTVECTOR\_67 Register (Offset = 210Ch) [Reset = 00000000h]

INTVECTOR\_67 is shown in [Table 13-401](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h71

**Table 13-401. INTVECTOR\_67 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.400 INTVECTOR\_68 Register (Offset = 2110h) [Reset = 00000000h]

INTVECTOR\_68 is shown in [Table 13-402](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h72

**Table 13-402. INTVECTOR\_68 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.401 INTVECTOR\_69 Register (Offset = 2114h) [Reset = 00000000h]

INTVECTOR\_69 is shown in [Table 13-403](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h73

**Table 13-403. INTVECTOR\_69 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.402 INTVECTOR\_70 Register (Offset = 2118h) [Reset = 00000000h]

INTVECTOR\_70 is shown in [Table 13-404](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h74

**Table 13-404. INTVECTOR\_70 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.403 INTVECTOR\_71 Register (Offset = 211Ch) [Reset = 00000000h]

INTVECTOR\_71 is shown in [Table 13-405](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h75

**Table 13-405. INTVECTOR\_71 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.404 INTVECTOR\_72 Register (Offset = 2120h) [Reset = 00000000h]

INTVECTOR\_72 is shown in [Table 13-406](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h76

**Table 13-406. INTVECTOR\_72 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.405 INTVECTOR\_73 Register (Offset = 2124h) [Reset = 00000000h]

INTVECTOR\_73 is shown in [Table 13-407](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h77

**Table 13-407. INTVECTOR\_73 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.406 INTVECTOR\_74 Register (Offset = 2128h) [Reset = 00000000h]

INTVECTOR\_74 is shown in [Table 13-408](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h78

**Table 13-408. INTVECTOR\_74 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.407 INTVECTOR\_75 Register (Offset = 212Ch) [Reset = 00000000h]

INTVECTOR\_75 is shown in [Table 13-409](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h79

**Table 13-409. INTVECTOR\_75 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.408 INTVECTOR\_76 Register (Offset = 2130h) [Reset = 00000000h]

INTVECTOR\_76 is shown in [Table 13-410](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h80

**Table 13-410. INTVECTOR\_76 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.409 INTVECTOR\_77 Register (Offset = 2134h) [Reset = 00000000h]

INTVECTOR\_77 is shown in [Table 13-411](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h81

**Table 13-411. INTVECTOR\_77 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.410 INTVECTOR\_78 Register (Offset = 2138h) [Reset = 00000000h]

INTVECTOR\_78 is shown in [Table 13-412](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h82

**Table 13-412. INTVECTOR\_78 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.411 INTVECTOR\_79 Register (Offset = 213Ch) [Reset = 00000000h]

INTVECTOR\_79 is shown in [Table 13-413](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h83

**Table 13-413. INTVECTOR\_79 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.412 INTVECTOR\_80 Register (Offset = 2140h) [Reset = 00000000h]

INTVECTOR\_80 is shown in [Table 13-414](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h84

**Table 13-414. INTVECTOR\_80 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.413 INTVECTOR\_81 Register (Offset = 2144h) [Reset = 00000000h]

INTVECTOR\_81 is shown in [Table 13-415](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h85

**Table 13-415. INTVECTOR\_81 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.414 INTVECTOR\_82 Register (Offset = 2148h) [Reset = 00000000h]

INTVECTOR\_82 is shown in [Table 13-416](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h86

**Table 13-416. INTVECTOR\_82 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.415 INTVECTOR\_83 Register (Offset = 214Ch) [Reset = 00000000h]

INTVECTOR\_83 is shown in [Table 13-417](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h87

**Table 13-417. INTVECTOR\_83 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.416 INTVECTOR\_84 Register (Offset = 2150h) [Reset = 00000000h]

INTVECTOR\_84 is shown in [Table 13-418](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h88

**Table 13-418. INTVECTOR\_84 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.417 INTVECTOR\_85 Register (Offset = 2154h) [Reset = 00000000h]

INTVECTOR\_85 is shown in [Table 13-419](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h89

**Table 13-419. INTVECTOR\_85 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.418 INTVECTOR\_86 Register (Offset = 2158h) [Reset = 00000000h]

INTVECTOR\_86 is shown in [Table 13-420](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h90

**Table 13-420. INTVECTOR\_86 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.419 INTVECTOR\_87 Register (Offset = 215Ch) [Reset = 00000000h]

INTVECTOR\_87 is shown in [Table 13-421](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h91

**Table 13-421. INTVECTOR\_87 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.420 INTVECTOR\_88 Register (Offset = 2160h) [Reset = 00000000h]

INTVECTOR\_88 is shown in [Table 13-422](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h92

**Table 13-422. INTVECTOR\_88 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.421 INTVECTOR\_89 Register (Offset = 2164h) [Reset = 00000000h]

INTVECTOR\_89 is shown in [Table 13-423](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h93

**Table 13-423. INTVECTOR\_89 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.422 INTVECTOR\_90 Register (Offset = 2168h) [Reset = 00000000h]

INTVECTOR\_90 is shown in [Table 13-424](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h94

**Table 13-424. INTVECTOR\_90 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.423 INTVECTOR\_91 Register (Offset = 216Ch) [Reset = 00000000h]

INTVECTOR\_91 is shown in [Table 13-425](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h95

**Table 13-425. INTVECTOR\_91 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.424 INTVECTOR\_92 Register (Offset = 2170h) [Reset = 00000000h]

INTVECTOR\_92 is shown in [Table 13-426](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h96

**Table 13-426. INTVECTOR\_92 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.425 INTVECTOR\_93 Register (Offset = 2174h) [Reset = 00000000h]

INTVECTOR\_93 is shown in [Table 13-427](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h97

**Table 13-427. INTVECTOR\_93 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.426 INTVECTOR\_94 Register (Offset = 2178h) [Reset = 00000000h]

INTVECTOR\_94 is shown in [Table 13-428](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h98

**Table 13-428. INTVECTOR\_94 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.427 INTVECTOR\_95 Register (Offset = 217Ch) [Reset = 00000000h]

INTVECTOR\_95 is shown in [Table 13-429](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h99

**Table 13-429. INTVECTOR\_95 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.428 INTVECTOR\_96 Register (Offset = 2180h) [Reset = 00000000h]

INTVECTOR\_96 is shown in [Table 13-430](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h100

**Table 13-430. INTVECTOR\_96 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.429 INTVECTOR\_97 Register (Offset = 2184h) [Reset = 00000000h]

INTVECTOR\_97 is shown in [Table 13-431](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h101

**Table 13-431. INTVECTOR\_97 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.430 INTVECTOR\_98 Register (Offset = 2188h) [Reset = 00000000h]

INTVECTOR\_98 is shown in [Table 13-432](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h102

**Table 13-432. INTVECTOR\_98 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.431 INTVECTOR\_99 Register (Offset = 218Ch) [Reset = 00000000h]

INTVECTOR\_99 is shown in [Table 13-433](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h103

**Table 13-433. INTVECTOR\_99 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.432 INTVECTOR\_100 Register (Offset = 2190h) [Reset = 00000000h]

INTVECTOR\_100 is shown in [Table 13-434](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h104

**Table 13-434. INTVECTOR\_100 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.433 INTVECTOR\_101 Register (Offset = 2194h) [Reset = 00000000h]

INTVECTOR\_101 is shown in [Table 13-435](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h105

**Table 13-435. INTVECTOR\_101 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.434 INTVECTOR\_102 Register (Offset = 2198h) [Reset = 00000000h]

INTVECTOR\_102 is shown in [Table 13-436](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h106

**Table 13-436. INTVECTOR\_102 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.435 INTVECTOR\_103 Register (Offset = 219Ch) [Reset = 00000000h]

INTVECTOR\_103 is shown in [Table 13-437](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h107

**Table 13-437. INTVECTOR\_103 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.436 INTVECTOR\_104 Register (Offset = 21A0h) [Reset = 00000000h]

INTVECTOR\_104 is shown in [Table 13-438](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h108

**Table 13-438. INTVECTOR\_104 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.437 INTVECTOR\_105 Register (Offset = 21A4h) [Reset = 00000000h]

INTVECTOR\_105 is shown in [Table 13-439](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h109

**Table 13-439. INTVECTOR\_105 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.438 INTVECTOR\_106 Register (Offset = 21A8h) [Reset = 00000000h]

INTVECTOR\_106 is shown in [Table 13-440](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h110

**Table 13-440. INTVECTOR\_106 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.439 INTVECTOR\_107 Register (Offset = 21ACh) [Reset = 00000000h]

INTVECTOR\_107 is shown in [Table 13-441](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h111

**Table 13-441. INTVECTOR\_107 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.440 INTVECTOR\_108 Register (Offset = 21B0h) [Reset = 00000000h]

INTVECTOR\_108 is shown in [Table 13-442](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h112

**Table 13-442. INTVECTOR\_108 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.441 INTVECTOR\_109 Register (Offset = 21B4h) [Reset = 00000000h]

INTVECTOR\_109 is shown in [Table 13-443](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h113

**Table 13-443. INTVECTOR\_109 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.442 INTVECTOR\_110 Register (Offset = 21B8h) [Reset = 00000000h]

INTVECTOR\_110 is shown in [Table 13-444](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h114

**Table 13-444. INTVECTOR\_110 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.443 INTVECTOR\_111 Register (Offset = 21BCh) [Reset = 00000000h]

INTVECTOR\_111 is shown in [Table 13-445](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h115

**Table 13-445. INTVECTOR\_111 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.444 INTVECTOR\_112 Register (Offset = 21C0h) [Reset = 00000000h]

INTVECTOR\_112 is shown in [Table 13-446](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h116

**Table 13-446. INTVECTOR\_112 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.445 INTVECTOR\_113 Register (Offset = 21C4h) [Reset = 00000000h]

INTVECTOR\_113 is shown in [Table 13-447](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h117

**Table 13-447. INTVECTOR\_113 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.446 INTVECTOR\_114 Register (Offset = 21C8h) [Reset = 00000000h]

INTVECTOR\_114 is shown in [Table 13-448](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h118

**Table 13-448. INTVECTOR\_114 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.447 INTVECTOR\_115 Register (Offset = 21CCh) [Reset = 00000000h]

INTVECTOR\_115 is shown in [Table 13-449](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h119

**Table 13-449. INTVECTOR\_115 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.448 INTVECTOR\_116 Register (Offset = 21D0h) [Reset = 00000000h]

INTVECTOR\_116 is shown in [Table 13-450](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h120

**Table 13-450. INTVECTOR\_116 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.449 INTVECTOR\_117 Register (Offset = 21D4h) [Reset = 00000000h]

INTVECTOR\_117 is shown in [Table 13-451](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h121

**Table 13-451. INTVECTOR\_117 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.450 INTVECTOR\_118 Register (Offset = 21D8h) [Reset = 00000000h]

INTVECTOR\_118 is shown in [Table 13-452](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h122

**Table 13-452. INTVECTOR\_118 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.451 INTVECTOR\_119 Register (Offset = 21DCh) [Reset = 00000000h]

INTVECTOR\_119 is shown in [Table 13-453](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h123

**Table 13-453. INTVECTOR\_119 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.452 INTVECTOR\_120 Register (Offset = 21E0h) [Reset = 00000000h]

INTVECTOR\_120 is shown in [Table 13-454](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h124

**Table 13-454. INTVECTOR\_120 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.453 INTVECTOR\_121 Register (Offset = 21E4h) [Reset = 00000000h]

INTVECTOR\_121 is shown in [Table 13-455](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h125

**Table 13-455. INTVECTOR\_121 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.454 INTVECTOR\_122 Register (Offset = 21E8h) [Reset = 00000000h]

INTVECTOR\_122 is shown in [Table 13-456](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h126

**Table 13-456. INTVECTOR\_122 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.455 INTVECTOR\_123 Register (Offset = 21ECh) [Reset = 00000000h]

INTVECTOR\_123 is shown in [Table 13-457](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h127

**Table 13-457. INTVECTOR\_123 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.456 INTVECTOR\_124 Register (Offset = 21F0h) [Reset = 00000000h]

INTVECTOR\_124 is shown in [Table 13-458](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h128

**Table 13-458. INTVECTOR\_124 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.457 INTVECTOR\_125 Register (Offset = 21F4h) [Reset = 00000000h]

INTVECTOR\_125 is shown in [Table 13-459](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h129

**Table 13-459. INTVECTOR\_125 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.458 INTVECTOR\_126 Register (Offset = 21F8h) [Reset = 00000000h]

INTVECTOR\_126 is shown in [Table 13-460](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h130

**Table 13-460. INTVECTOR\_126 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.459 INTVECTOR\_127 Register (Offset = 21FCh) [Reset = 00000000h]

INTVECTOR\_127 is shown in [Table 13-461](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h131

**Table 13-461. INTVECTOR\_127 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.460 INTVECTOR\_128 Register (Offset = 2200h) [Reset = 00000000h]

INTVECTOR\_128 is shown in [Table 13-462](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h132

**Table 13-462. INTVECTOR\_128 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.461 INTVECTOR\_129 Register (Offset = 2204h) [Reset = 00000000h]

INTVECTOR\_129 is shown in [Table 13-463](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h133

**Table 13-463. INTVECTOR\_129 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.462 INTVECTOR\_130 Register (Offset = 2208h) [Reset = 00000000h]

INTVECTOR\_130 is shown in [Table 13-464](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h134

**Table 13-464. INTVECTOR\_130 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.463 INTVECTOR\_131 Register (Offset = 220Ch) [Reset = 00000000h]

INTVECTOR\_131 is shown in [Table 13-465](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h135

**Table 13-465. INTVECTOR\_131 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.464 INTVECTOR\_132 Register (Offset = 2210h) [Reset = 00000000h]

INTVECTOR\_132 is shown in [Table 13-466](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h136

**Table 13-466. INTVECTOR\_132 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.465 INTVECTOR\_133 Register (Offset = 2214h) [Reset = 00000000h]

INTVECTOR\_133 is shown in [Table 13-467](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h137

**Table 13-467. INTVECTOR\_133 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.466 INTVECTOR\_134 Register (Offset = 2218h) [Reset = 00000000h]

INTVECTOR\_134 is shown in [Table 13-468](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h138

**Table 13-468. INTVECTOR\_134 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.467 INTVECTOR\_135 Register (Offset = 221Ch) [Reset = 00000000h]

INTVECTOR\_135 is shown in [Table 13-469](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h139

**Table 13-469. INTVECTOR\_135 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.468 INTVECTOR\_136 Register (Offset = 2220h) [Reset = 00000000h]

INTVECTOR\_136 is shown in [Table 13-470](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h140

**Table 13-470. INTVECTOR\_136 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.469 INTVECTOR\_137 Register (Offset = 2224h) [Reset = 00000000h]

INTVECTOR\_137 is shown in [Table 13-471](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h141

**Table 13-471. INTVECTOR\_137 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.470 INTVECTOR\_138 Register (Offset = 2228h) [Reset = 00000000h]

INTVECTOR\_138 is shown in [Table 13-472](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h142

**Table 13-472. INTVECTOR\_138 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.471 INTVECTOR\_139 Register (Offset = 222Ch) [Reset = 00000000h]

INTVECTOR\_139 is shown in [Table 13-473](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h143

**Table 13-473. INTVECTOR\_139 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.472 INTVECTOR\_140 Register (Offset = 2230h) [Reset = 00000000h]

INTVECTOR\_140 is shown in [Table 13-474](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h144

**Table 13-474. INTVECTOR\_140 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.473 INTVECTOR\_141 Register (Offset = 2234h) [Reset = 00000000h]

INTVECTOR\_141 is shown in [Table 13-475](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h145

**Table 13-475. INTVECTOR\_141 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.474 INTVECTOR\_142 Register (Offset = 2238h) [Reset = 00000000h]

INTVECTOR\_142 is shown in [Table 13-476](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h146

**Table 13-476. INTVECTOR\_142 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.475 INTVECTOR\_143 Register (Offset = 223Ch) [Reset = 00000000h]

INTVECTOR\_143 is shown in [Table 13-477](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h147

**Table 13-477. INTVECTOR\_143 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.476 INTVECTOR\_144 Register (Offset = 2240h) [Reset = 00000000h]

INTVECTOR\_144 is shown in [Table 13-478](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h148

**Table 13-478. INTVECTOR\_144 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.477 INTVECTOR\_145 Register (Offset = 2244h) [Reset = 00000000h]

INTVECTOR\_145 is shown in [Table 13-479](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h149

**Table 13-479. INTVECTOR\_145 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.478 INTVECTOR\_146 Register (Offset = 2248h) [Reset = 00000000h]

INTVECTOR\_146 is shown in [Table 13-480](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h150

**Table 13-480. INTVECTOR\_146 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.479 INTVECTOR\_147 Register (Offset = 224Ch) [Reset = 00000000h]

INTVECTOR\_147 is shown in [Table 13-481](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h151

**Table 13-481. INTVECTOR\_147 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.480 INTVECTOR\_148 Register (Offset = 2250h) [Reset = 00000000h]

INTVECTOR\_148 is shown in [Table 13-482](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h152

**Table 13-482. INTVECTOR\_148 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.481 INTVECTOR\_149 Register (Offset = 2254h) [Reset = 00000000h]

INTVECTOR\_149 is shown in [Table 13-483](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h153

**Table 13-483. INTVECTOR\_149 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.482 INTVECTOR\_150 Register (Offset = 2258h) [Reset = 00000000h]

INTVECTOR\_150 is shown in [Table 13-484](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h154

**Table 13-484. INTVECTOR\_150 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.483 INTVECTOR\_151 Register (Offset = 225Ch) [Reset = 00000000h]

INTVECTOR\_151 is shown in [Table 13-485](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h155

**Table 13-485. INTVECTOR\_151 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.484 INTVECTOR\_152 Register (Offset = 2260h) [Reset = 00000000h]

INTVECTOR\_152 is shown in [Table 13-486](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h156

**Table 13-486. INTVECTOR\_152 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.485 INTVECTOR\_153 Register (Offset = 2264h) [Reset = 00000000h]

INTVECTOR\_153 is shown in [Table 13-487](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h157

**Table 13-487. INTVECTOR\_153 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.486 INTVECTOR\_154 Register (Offset = 2268h) [Reset = 00000000h]

INTVECTOR\_154 is shown in [Table 13-488](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h158

**Table 13-488. INTVECTOR\_154 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.487 INTVECTOR\_155 Register (Offset = 226Ch) [Reset = 00000000h]

INTVECTOR\_155 is shown in [Table 13-489](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h159

**Table 13-489. INTVECTOR\_155 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.488 INTVECTOR\_156 Register (Offset = 2270h) [Reset = 00000000h]

INTVECTOR\_156 is shown in [Table 13-490](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h160

**Table 13-490. INTVECTOR\_156 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.489 INTVECTOR\_157 Register (Offset = 2274h) [Reset = 00000000h]

INTVECTOR\_157 is shown in [Table 13-491](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h161

**Table 13-491. INTVECTOR\_157 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.490 INTVECTOR\_158 Register (Offset = 2278h) [Reset = 00000000h]

INTVECTOR\_158 is shown in [Table 13-492](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h162

**Table 13-492. INTVECTOR\_158 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.491 INTVECTOR\_159 Register (Offset = 227Ch) [Reset = 00000000h]

INTVECTOR\_159 is shown in [Table 13-493](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h163

**Table 13-493. INTVECTOR\_159 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.492 INTVECTOR\_160 Register (Offset = 2280h) [Reset = 00000000h]

INTVECTOR\_160 is shown in [Table 13-494](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h164

**Table 13-494. INTVECTOR\_160 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.493 INTVECTOR\_161 Register (Offset = 2284h) [Reset = 00000000h]

INTVECTOR\_161 is shown in [Table 13-495](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h165

**Table 13-495. INTVECTOR\_161 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.494 INTVECTOR\_162 Register (Offset = 2288h) [Reset = 00000000h]

INTVECTOR\_162 is shown in [Table 13-496](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h166

**Table 13-496. INTVECTOR\_162 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.495 INTVECTOR\_163 Register (Offset = 228Ch) [Reset = 00000000h]

INTVECTOR\_163 is shown in [Table 13-497](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h167

**Table 13-497. INTVECTOR\_163 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.496 INTVECTOR\_164 Register (Offset = 2290h) [Reset = 00000000h]

INTVECTOR\_164 is shown in [Table 13-498](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h168

**Table 13-498. INTVECTOR\_164 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.497 INTVECTOR\_165 Register (Offset = 2294h) [Reset = 00000000h]

INTVECTOR\_165 is shown in [Table 13-499](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h169

**Table 13-499. INTVECTOR\_165 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.498 INTVECTOR\_166 Register (Offset = 2298h) [Reset = 00000000h]

INTVECTOR\_166 is shown in [Table 13-500](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h170

**Table 13-500. INTVECTOR\_166 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.499 INTVECTOR\_167 Register (Offset = 229Ch) [Reset = 00000000h]

INTVECTOR\_167 is shown in [Table 13-501](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h171

**Table 13-501. INTVECTOR\_167 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.500 INTVECTOR\_168 Register (Offset = 22A0h) [Reset = 00000000h]

INTVECTOR\_168 is shown in [Table 13-502](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h172

**Table 13-502. INTVECTOR\_168 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.501 INTVECTOR\_169 Register (Offset = 22A4h) [Reset = 00000000h]

INTVECTOR\_169 is shown in [Table 13-503](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h173

**Table 13-503. INTVECTOR\_169 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.502 INTVECTOR\_170 Register (Offset = 22A8h) [Reset = 00000000h]

INTVECTOR\_170 is shown in [Table 13-504](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h174

**Table 13-504. INTVECTOR\_170 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.503 INTVECTOR\_171 Register (Offset = 22ACh) [Reset = 00000000h]

INTVECTOR\_171 is shown in [Table 13-505](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h175

**Table 13-505. INTVECTOR\_171 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.504 INTVECTOR\_172 Register (Offset = 22B0h) [Reset = 00000000h]

INTVECTOR\_172 is shown in [Table 13-506](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h176

**Table 13-506. INTVECTOR\_172 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.505 INTVECTOR\_173 Register (Offset = 22B4h) [Reset = 00000000h]

INTVECTOR\_173 is shown in [Table 13-507](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h177

**Table 13-507. INTVECTOR\_173 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.506 INTVECTOR\_174 Register (Offset = 22B8h) [Reset = 00000000h]

INTVECTOR\_174 is shown in [Table 13-508](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h178

**Table 13-508. INTVECTOR\_174 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.507 INTVECTOR\_175 Register (Offset = 22BCh) [Reset = 00000000h]

INTVECTOR\_175 is shown in [Table 13-509](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h179

**Table 13-509. INTVECTOR\_175 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.508 INTVECTOR\_176 Register (Offset = 22C0h) [Reset = 00000000h]

INTVECTOR\_176 is shown in [Table 13-510](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h180

**Table 13-510. INTVECTOR\_176 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.509 INTVECTOR\_177 Register (Offset = 22C4h) [Reset = 00000000h]

INTVECTOR\_177 is shown in [Table 13-511](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h181

**Table 13-511. INTVECTOR\_177 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.510 INTVECTOR\_178 Register (Offset = 22C8h) [Reset = 00000000h]

INTVECTOR\_178 is shown in [Table 13-512](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h182

**Table 13-512. INTVECTOR\_178 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.511 INTVECTOR\_179 Register (Offset = 22CCh) [Reset = 00000000h]

INTVECTOR\_179 is shown in [Table 13-513](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h183

**Table 13-513. INTVECTOR\_179 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.512 INTVECTOR\_180 Register (Offset = 22D0h) [Reset = 00000000h]

INTVECTOR\_180 is shown in [Table 13-514](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h184

**Table 13-514. INTVECTOR\_180 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.513 INTVECTOR\_181 Register (Offset = 22D4h) [Reset = 00000000h]

INTVECTOR\_181 is shown in [Table 13-515](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h185

**Table 13-515. INTVECTOR\_181 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.514 INTVECTOR\_182 Register (Offset = 22D8h) [Reset = 00000000h]

INTVECTOR\_182 is shown in [Table 13-516](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h186

**Table 13-516. INTVECTOR\_182 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.515 INTVECTOR\_183 Register (Offset = 22DCh) [Reset = 00000000h]

INTVECTOR\_183 is shown in [Table 13-517](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h187

**Table 13-517. INTVECTOR\_183 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.516 INTVECTOR\_184 Register (Offset = 22E0h) [Reset = 00000000h]

INTVECTOR\_184 is shown in [Table 13-518](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h188

**Table 13-518. INTVECTOR\_184 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.517 INTVECTOR\_185 Register (Offset = 22E4h) [Reset = 00000000h]

INTVECTOR\_185 is shown in [Table 13-519](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h189

**Table 13-519. INTVECTOR\_185 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.518 INTVECTOR\_186 Register (Offset = 22E8h) [Reset = 00000000h]

INTVECTOR\_186 is shown in [Table 13-520](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h190

**Table 13-520. INTVECTOR\_186 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.519 INTVECTOR\_187 Register (Offset = 22ECh) [Reset = 00000000h]

INTVECTOR\_187 is shown in [Table 13-521](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h191

**Table 13-521. INTVECTOR\_187 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.520 INTVECTOR\_188 Register (Offset = 22F0h) [Reset = 00000000h]

INTVECTOR\_188 is shown in [Table 13-522](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h192

**Table 13-522. INTVECTOR\_188 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.521 INTVECTOR\_189 Register (Offset = 22F4h) [Reset = 00000000h]

INTVECTOR\_189 is shown in [Table 13-523](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h193

**Table 13-523. INTVECTOR\_189 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.522 INTVECTOR\_190 Register (Offset = 22F8h) [Reset = 00000000h]

INTVECTOR\_190 is shown in [Table 13-524](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h194

**Table 13-524. INTVECTOR\_190 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.523 INTVECTOR\_191 Register (Offset = 22FCh) [Reset = 00000000h]

INTVECTOR\_191 is shown in [Table 13-525](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h195

**Table 13-525. INTVECTOR\_191 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.524 INTVECTOR\_192 Register (Offset = 2300h) [Reset = 00000000h]

INTVECTOR\_192 is shown in [Table 13-526](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h196

**Table 13-526. INTVECTOR\_192 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.525 INTVECTOR\_193 Register (Offset = 2304h) [Reset = 00000000h]

INTVECTOR\_193 is shown in [Table 13-527](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h197

**Table 13-527. INTVECTOR\_193 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.526 INTVECTOR\_194 Register (Offset = 2308h) [Reset = 00000000h]

INTVECTOR\_194 is shown in [Table 13-528](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h198

**Table 13-528. INTVECTOR\_194 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.527 INTVECTOR\_195 Register (Offset = 230Ch) [Reset = 00000000h]

INTVECTOR\_195 is shown in [Table 13-529](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h199

**Table 13-529. INTVECTOR\_195 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.528 INTVECTOR\_196 Register (Offset = 2310h) [Reset = 00000000h]

INTVECTOR\_196 is shown in [Table 13-530](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h200

**Table 13-530. INTVECTOR\_196 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.529 INTVECTOR\_197 Register (Offset = 2314h) [Reset = 00000000h]

INTVECTOR\_197 is shown in [Table 13-531](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h201

**Table 13-531. INTVECTOR\_197 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.530 INTVECTOR\_198 Register (Offset = 2318h) [Reset = 00000000h]

INTVECTOR\_198 is shown in [Table 13-532](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h202

**Table 13-532. INTVECTOR\_198 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.531 INTVECTOR\_199 Register (Offset = 231Ch) [Reset = 00000000h]

INTVECTOR\_199 is shown in [Table 13-533](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h203

**Table 13-533. INTVECTOR\_199 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.532 INTVECTOR\_200 Register (Offset = 2320h) [Reset = 00000000h]

INTVECTOR\_200 is shown in [Table 13-534](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h204

**Table 13-534. INTVECTOR\_200 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.533 INTVECTOR\_201 Register (Offset = 2324h) [Reset = 00000000h]

INTVECTOR\_201 is shown in [Table 13-535](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h205

**Table 13-535. INTVECTOR\_201 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.534 INTVECTOR\_202 Register (Offset = 2328h) [Reset = 00000000h]

INTVECTOR\_202 is shown in [Table 13-536](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h206

**Table 13-536. INTVECTOR\_202 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.535 INTVECTOR\_203 Register (Offset = 232Ch) [Reset = 00000000h]

INTVECTOR\_203 is shown in [Table 13-537](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h207

**Table 13-537. INTVECTOR\_203 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.536 INTVECTOR\_204 Register (Offset = 2330h) [Reset = 00000000h]

INTVECTOR\_204 is shown in [Table 13-538](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h208

**Table 13-538. INTVECTOR\_204 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.537 INTVECTOR\_205 Register (Offset = 2334h) [Reset = 00000000h]

INTVECTOR\_205 is shown in [Table 13-539](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h209

**Table 13-539. INTVECTOR\_205 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.538 INTVECTOR\_206 Register (Offset = 2338h) [Reset = 00000000h]

INTVECTOR\_206 is shown in [Table 13-540](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h210

**Table 13-540. INTVECTOR\_206 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.539 INTVECTOR\_207 Register (Offset = 233Ch) [Reset = 00000000h]

INTVECTOR\_207 is shown in [Table 13-541](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h211

**Table 13-541. INTVECTOR\_207 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.540 INTVECTOR\_208 Register (Offset = 2340h) [Reset = 00000000h]

INTVECTOR\_208 is shown in [Table 13-542](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h212

**Table 13-542. INTVECTOR\_208 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.541 INTVECTOR\_209 Register (Offset = 2344h) [Reset = 00000000h]

INTVECTOR\_209 is shown in [Table 13-543](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h213

**Table 13-543. INTVECTOR\_209 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.542 INTVECTOR\_210 Register (Offset = 2348h) [Reset = 00000000h]

INTVECTOR\_210 is shown in [Table 13-544](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h214

**Table 13-544. INTVECTOR\_210 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.543 INTVECTOR\_211 Register (Offset = 234Ch) [Reset = 00000000h]

INTVECTOR\_211 is shown in [Table 13-545](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h215

**Table 13-545. INTVECTOR\_211 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.544 INTVECTOR\_212 Register (Offset = 2350h) [Reset = 00000000h]

INTVECTOR\_212 is shown in [Table 13-546](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h216

**Table 13-546. INTVECTOR\_212 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.545 INTVECTOR\_213 Register (Offset = 2354h) [Reset = 00000000h]

INTVECTOR\_213 is shown in [Table 13-547](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h217

**Table 13-547. INTVECTOR\_213 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.546 INTVECTOR\_214 Register (Offset = 2358h) [Reset = 00000000h]

INTVECTOR\_214 is shown in [Table 13-548](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h218

**Table 13-548. INTVECTOR\_214 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.547 INTVECTOR\_215 Register (Offset = 235Ch) [Reset = 00000000h]

INTVECTOR\_215 is shown in [Table 13-549](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h219

**Table 13-549. INTVECTOR\_215 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.548 INTVECTOR\_216 Register (Offset = 2360h) [Reset = 00000000h]

INTVECTOR\_216 is shown in [Table 13-550](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h220

**Table 13-550. INTVECTOR\_216 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.549 INTVECTOR\_217 Register (Offset = 2364h) [Reset = 00000000h]

INTVECTOR\_217 is shown in [Table 13-551](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h221

**Table 13-551. INTVECTOR\_217 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.550 INTVECTOR\_218 Register (Offset = 2368h) [Reset = 00000000h]

INTVECTOR\_218 is shown in [Table 13-552](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h222

**Table 13-552. INTVECTOR\_218 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.551 INTVECTOR\_219 Register (Offset = 236Ch) [Reset = 00000000h]

INTVECTOR\_219 is shown in [Table 13-553](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h223

**Table 13-553. INTVECTOR\_219 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.552 INTVECTOR\_220 Register (Offset = 2370h) [Reset = 00000000h]

INTVECTOR\_220 is shown in [Table 13-554](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h224

**Table 13-554. INTVECTOR\_220 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.553 INTVECTOR\_221 Register (Offset = 2374h) [Reset = 00000000h]

INTVECTOR\_221 is shown in [Table 13-555](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h225

**Table 13-555. INTVECTOR\_221 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.554 INTVECTOR\_222 Register (Offset = 2378h) [Reset = 00000000h]

INTVECTOR\_222 is shown in [Table 13-556](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h226

**Table 13-556. INTVECTOR\_222 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.555 INTVECTOR\_223 Register (Offset = 237Ch) [Reset = 00000000h]

INTVECTOR\_223 is shown in [Table 13-557](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h227

**Table 13-557. INTVECTOR\_223 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.556 INTVECTOR\_224 Register (Offset = 2380h) [Reset = 00000000h]

INTVECTOR\_224 is shown in [Table 13-558](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h228

**Table 13-558. INTVECTOR\_224 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.557 INTVECTOR\_225 Register (Offset = 2384h) [Reset = 00000000h]

INTVECTOR\_225 is shown in [Table 13-559](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h229

**Table 13-559. INTVECTOR\_225 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.558 INTVECTOR\_226 Register (Offset = 2388h) [Reset = 00000000h]

INTVECTOR\_226 is shown in [Table 13-560](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h230

**Table 13-560. INTVECTOR\_226 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.559 INTVECTOR\_227 Register (Offset = 238Ch) [Reset = 00000000h]

INTVECTOR\_227 is shown in [Table 13-561](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h231

**Table 13-561. INTVECTOR\_227 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.560 INTVECTOR\_228 Register (Offset = 2390h) [Reset = 00000000h]

INTVECTOR\_228 is shown in [Table 13-562](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h232

**Table 13-562. INTVECTOR\_228 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.561 INTVECTOR\_229 Register (Offset = 2394h) [Reset = 00000000h]

INTVECTOR\_229 is shown in [Table 13-563](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h233

**Table 13-563. INTVECTOR\_229 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.562 INTVECTOR\_230 Register (Offset = 2398h) [Reset = 00000000h]

INTVECTOR\_230 is shown in [Table 13-564](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h234

**Table 13-564. INTVECTOR\_230 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.563 INTVECTOR\_231 Register (Offset = 239Ch) [Reset = 00000000h]

INTVECTOR\_231 is shown in [Table 13-565](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h235

**Table 13-565. INTVECTOR\_231 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.564 INTVECTOR\_232 Register (Offset = 23A0h) [Reset = 00000000h]

INTVECTOR\_232 is shown in [Table 13-566](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h236

**Table 13-566. INTVECTOR\_232 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.565 INTVECTOR\_233 Register (Offset = 23A4h) [Reset = 00000000h]

INTVECTOR\_233 is shown in [Table 13-567](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h237

**Table 13-567. INTVECTOR\_233 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.566 INTVECTOR\_234 Register (Offset = 23A8h) [Reset = 00000000h]

INTVECTOR\_234 is shown in [Table 13-568](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h238

**Table 13-568. INTVECTOR\_234 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.567 INTVECTOR\_235 Register (Offset = 23ACh) [Reset = 00000000h]

INTVECTOR\_235 is shown in [Table 13-569](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h239

**Table 13-569. INTVECTOR\_235 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.568 INTVECTOR\_236 Register (Offset = 23B0h) [Reset = 00000000h]

INTVECTOR\_236 is shown in [Table 13-570](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h240

**Table 13-570. INTVECTOR\_236 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.569 INTVECTOR\_237 Register (Offset = 23B4h) [Reset = 00000000h]

INTVECTOR\_237 is shown in [Table 13-571](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h241

**Table 13-571. INTVECTOR\_237 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.570 INTVECTOR\_238 Register (Offset = 23B8h) [Reset = 00000000h]

INTVECTOR\_238 is shown in [Table 13-572](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h242

**Table 13-572. INTVECTOR\_238 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.571 INTVECTOR\_239 Register (Offset = 23BCh) [Reset = 00000000h]

INTVECTOR\_239 is shown in [Table 13-573](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h243

**Table 13-573. INTVECTOR\_239 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.572 INTVECTOR\_240 Register (Offset = 23C0h) [Reset = 00000000h]

INTVECTOR\_240 is shown in [Table 13-574](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h244

**Table 13-574. INTVECTOR\_240 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.573 INTVECTOR\_241 Register (Offset = 23C4h) [Reset = 00000000h]

INTVECTOR\_241 is shown in [Table 13-575](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h245

**Table 13-575. INTVECTOR\_241 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.574 INTVECTOR\_242 Register (Offset = 23C8h) [Reset = 00000000h]

INTVECTOR\_242 is shown in [Table 13-576](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h246

**Table 13-576. INTVECTOR\_242 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.575 INTVECTOR\_243 Register (Offset = 23CCh) [Reset = 00000000h]

INTVECTOR\_243 is shown in [Table 13-577](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h247

**Table 13-577. INTVECTOR\_243 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.576 INTVECTOR\_244 Register (Offset = 23D0h) [Reset = 00000000h]

INTVECTOR\_244 is shown in [Table 13-578](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h248

**Table 13-578. INTVECTOR\_244 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.577 INTVECTOR\_245 Register (Offset = 23D4h) [Reset = 00000000h]

INTVECTOR\_245 is shown in [Table 13-579](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h249

**Table 13-579. INTVECTOR\_245 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.578 INTVECTOR\_246 Register (Offset = 23D8h) [Reset = 00000000h]

INTVECTOR\_246 is shown in [Table 13-580](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h250

**Table 13-580. INTVECTOR\_246 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.579 INTVECTOR\_247 Register (Offset = 23DCh) [Reset = 00000000h]

INTVECTOR\_247 is shown in [Table 13-581](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h251

**Table 13-581. INTVECTOR\_247 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.580 INTVECTOR\_248 Register (Offset = 23E0h) [Reset = 00000000h]

INTVECTOR\_248 is shown in [Table 13-582](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h252

**Table 13-582. INTVECTOR\_248 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.581 INTVECTOR\_249 Register (Offset = 23E4h) [Reset = 00000000h]

INTVECTOR\_249 is shown in [Table 13-583](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h253

**Table 13-583. INTVECTOR\_249 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>



### 13.10.582 INTVECTOR\_250 Register (Offset = 23E8h) [Reset = 00000000h]

INTVECTOR\_250 is shown in [Table 13-584](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h254

**Table 13-584. INTVECTOR\_250 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.583 INTVECTOR\_251 Register (Offset = 23ECh) [Reset = 00000000h]

INTVECTOR\_251 is shown in [Table 13-585](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h255

**Table 13-585. INTVECTOR\_251 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.584 INTVECTOR\_252 Register (Offset = 23F0h) [Reset = 00000000h]

INTVECTOR\_252 is shown in [Table 13-586](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h256

**Table 13-586. INTVECTOR\_252 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.585 INTVECTOR\_253 Register (Offset = 23F4h) [Reset = 00000000h]

INTVECTOR\_253 is shown in [Table 13-587](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h257

**Table 13-587. INTVECTOR\_253 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.586 INTVECTOR\_254 Register (Offset = 23F8h) [Reset = 00000000h]

INTVECTOR\_254 is shown in [Table 13-588](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h258

**Table 13-588. INTVECTOR\_254 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

### 13.10.587 INTVECTOR\_255 Register (Offset = 23FCh) [Reset = 00000000h]

INTVECTOR\_255 is shown in [Table 13-589](#).

Return to the [Summary Table](#).

Interrupt Q Vector Register (Q is 0 to 255 , Q= M+1 x 32) h2000 + Q x h259

**Table 13-589. INTVECTOR\_255 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	ADDR	R/W	0h	<p>This is the 32-bit Vector Address associated with interrupt Q. It is the address that will be reflected in the IRQ Vector Address (Base Address + 0x18) or FIQ Vector Address (Base Address + 0x1C) and the VECADDR pin when interrupt Q is the active interrupt. Internally, these values are kept in a RAM. The FIQ and IRQ state machines have priority access to this RAM. Writes to this register will be piped internally, but further writes to the MMR interface may be stalled until this write has a chance to complete in the RAM. The new Vector Address will not take effect until this write completes to the RAM. In order to tell if this write has completed, software may read this register back. That read will not be able to complete unless the write has landed. Reads to this register will stall the MMR interface until the read is able to be completed at the RAM.</p>
1-0	RES20	R	0h	<p>Reserved. Read as 0. The lower 2 bits of the 32-bit vector address are always 0. Vector addresses must be 32-bit aligned.</p>

## Chapter 14

# Real-Time Interrupt (RTI) and Watchdog Module

---



This chapter describes the functionality of the real-time interrupt (RTI) module. The RTI is designed as an operating system timer to support a real time operating system (RTOS).

---

### Note

This chapter describes a superset implementation of the RTI module that includes features and functionality related to DMA and Timebase control. These features are dependent on the device-specific feature content. Consult your device-specific datasheet to determine the applicability of these features to your device being used.

---

## 14.1 Overview

The real-time interrupt (RTI) module provides timer functionality for operating systems and for benchmarking code. The RTI module can incorporate several counters that define the time bases needed for scheduling in the operating system.

The timers also allow you to benchmark certain areas of code by reading the values of the counters at the beginning and the end of the desired code range and calculating the difference between the values.

There are two instances of the RTI and Watchdog modules: One in the APPSS, and one in the HWASS.

### 14.1.1 Features

The RTI module has the following features:

- Two independent 64 bit counter blocks
- Windowed Watchdog Timer (WWDT) Feature
- Four configurable compares for generating operating system ticks or DMA requests. Each event can be driven by either counter block 0 or counter block 1.
- Fast enabling/disabling of events
- Two time stamp (capture) functions for system or peripheral interrupts, one for each counter block
- Digital windowed watchdog

The RTI does not support the following features:

- External clock supervising circuit to switch to internal prescale counter 0, if external clock source fails to increment in a predefined window.
- Capture events to capture timestamps through recording of timer status.
- Two time-stamp (capture) functions for system or peripheral interrupts, one for each counter block.
- Analog Watchdog via external RC Network to prevent for runaway code.

### 14.1.2 Industry Standard Compliance Statement

This module is specifically designed to fulfill the requirements for OSEK (**O**ffene **S**ysteme und deren **S**chnittstellen für die **E**lektronik im **K**raftfahrzeug, or Open Systems and the Corresponding Interfaces for Automotive Electronics) as well as OSEK/time-compliant operating systems, but is not limited to it.

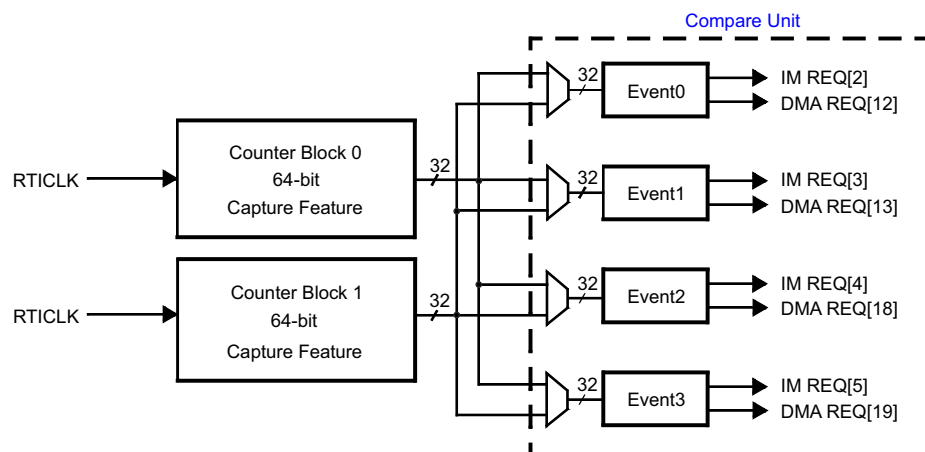
## 14.2 Module Operation

Figure 14-1 illustrates the high level block diagram of the RTI module.

The RTI module has two independent counter blocks for generating different timebases: counter block 0 and counter block 1. The two counter blocks provide the same basic functionality.

A compare unit compares the counters with programmable values and generates four independent interrupt or DMA requests on compare matches. Each of the compare registers can be programmed to be compared to either counter block 0 or counter block 1.

The following sections describe the individual functions in more detail.



**Figure 14-1. RTI Block Diagram**

### Note

During deep sleep entry, there is a need to disable the WDT warm reset propagation.

The WDT does **not** provide coverage in deep sleep mode. Thus, proper care must be taken by the application to ensure the WDT is not enabled when entering this mode.

TOP\_PRCM:RST\_WDT\_RESET\_EN[0] must be set to 0.

### 14.2.1 Counter Operation

Each counter block consists of the following (see Figure 14-2):

- One 32-bit prescale counter (RTIUC0 or RTIUC1)
- One 32-bit free running counter (RTIFRC0 or RTIFRC1)

The RTIUC0/1 is driven by the RTICLK and counts up until the compare value in the compare up counter register (RTICPUC0 or RTICPUC1) is reached. When the compare matches, RTIFRC0/1 is incremented and RTIUC0/1 is reset to 0. If RTIFRC0/1 overflows, an interrupt is generated to the interrupt manager (NVIC/IM). The overflow interrupt is not intended to generate the timebase for the operating system. See Section 14.2.2 for the timebase generation. The up counter together with the compare up counter value prescale the RTI clock. The resulting formula for the frequency of the free running counter (RTIFRC0/1) is:

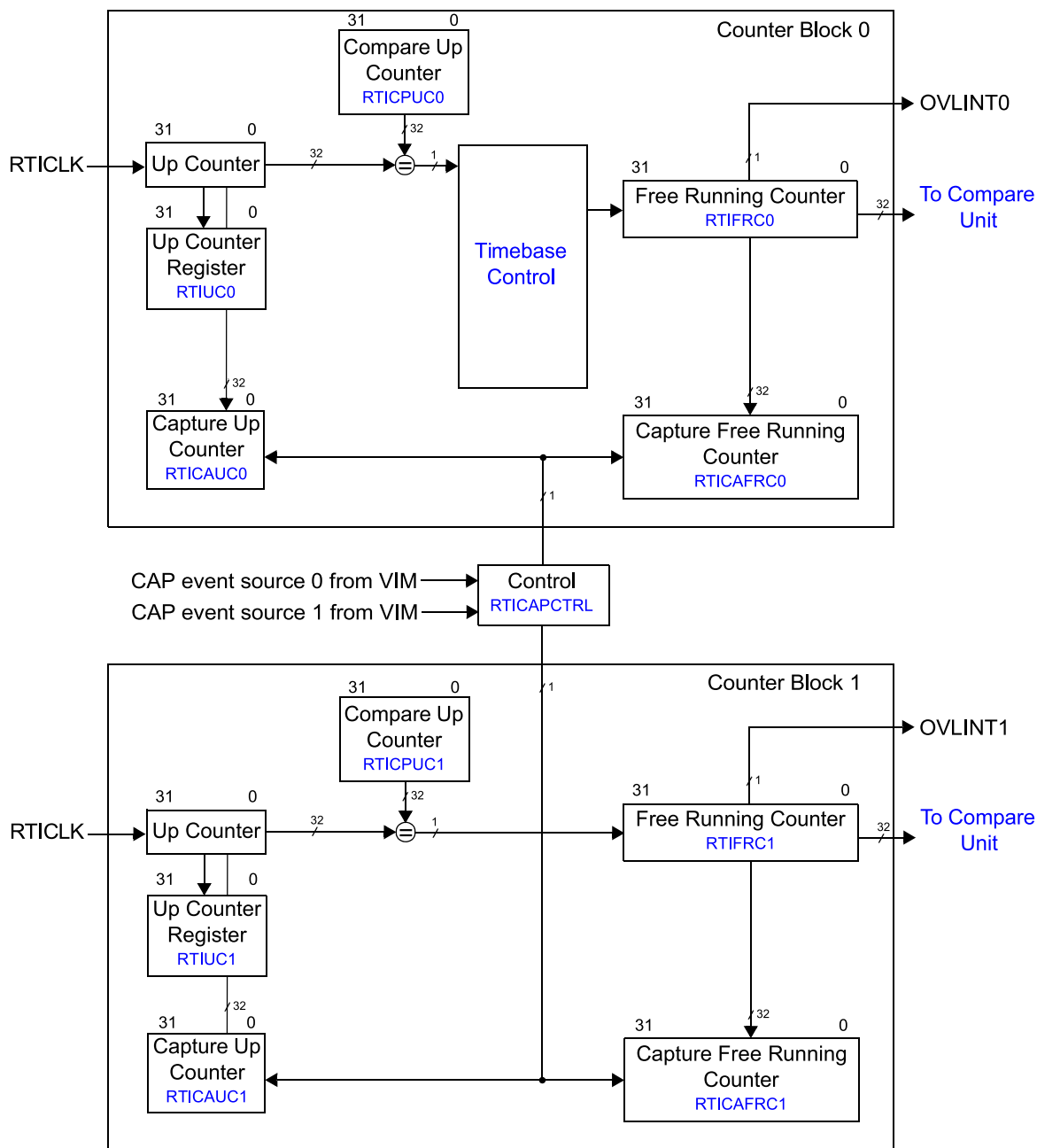
$$f_{RTIFRCx} = \begin{cases} \frac{f_{RTICLK}}{RTICPUCx + 1} & \text{when } RTICPUCx \neq 0 \\ \frac{f_{RTICLK}}{2^{32} + 1} & \text{when } RTICPUCx = 0 \end{cases} \quad (1)$$



### Note

Setting RTICPUCx equal to zero is not recommended. Doing so will hold the Up Counter at zero for two RTICLK cycles after it overflows from 0xFFFFFFFF to zero.

The counter values can be determined by reading the respective counter registers or by generating a hardware event which captures the counter value into the respective capture register. Both functions are described in the following sections.



**Figure 14-2. Counter Block Diagram**

#### 14.2.1.1 Counter and Capture Read Consistency

Portions of the device internal databus are 32-bits wide. If the application wants to read the 64-bit counters or the 64-bit capture values, a certain order of 32-bit read operations needs to be followed. This is to prevent one counter incrementing in between the two separate read operations to both counters.

##### Reading the Counters

The free running counter (RTIFRCx) must be read first. This priority will ensure that in the cycle when the CPU reads RTIFRCx, the up counter value is stored in its counter register (RTIUCx). The second read has to access the up counter register (RTIUCx), which then holds the value which corresponds to the number of RTICLK cycles that have elapsed at the time reading the free running counter register (RTIFRCx).

---

##### Note

The up counters are implemented as shadow registers. Reading RTIUCx without having read RTIFRCx first will return always the same value. RTIUCx will only be updated when RTIFRCx is read.

---

##### Reading the Capture Values

The free running counter capture register (RTICAFRCx) must be read first. This priority will ensure that in the cycle when the CPU reads RTICAFRCx, the up counter value is stored in its counter register (RTICAUCx). The second read has to access the up counter register (RTICAUCx), which then holds the value captured at the time when reading the capture free running counter register (RTICAFRCx).

---

##### Note

The capture up counter registers are implemented as shadow registers. Reading RTICAUCx without having read RTICAFRCx first will return always the same value. RTICAUCx will only be updated when RTICAFRCx is read.

---

#### 14.2.1.2 Capture Feature

Both counter blocks also provide a capture feature on external events. Two capture sources can trigger the capture event. The source triggering the block is configurable (RTICAPCTRL). The sources originate from the Interrupt Manager (NVIC/IM) and allow the generation of capture events when a peripheral modules has generated an interrupt. Any of the peripheral interrupts can be selected as the capture event in the NVIC/IM.

When an event is detected, RTIUCx and RTIFRCx are stored in the capture up counter (RTICAUCx) and capture free running counter (RTICAFRCx) registers. The read order of the captured values must be the same as the read order of the actual counters (see [Section 14.2.1.1](#)).

#### 14.2.2 Interrupt/DMA Requests

There are four compare registers (RTICOMPy) to generate interrupt requests to the NVIC/IM or DMA requests to the DMA controller. The interrupts can be used to generate different timebases for the operating system. Each of the compare registers can be configured to be compared to either RTIFRC0 or RTIFRC1. When the counter value matches the compare value, an interrupt is generated. To allow periodic interrupts, a certain value can be added to the compare value in RTICOMPy automatically. This value is stored in the update compare register (RTIUDCPy) and will be added after a compare is matched. The period of the generated interrupt/DMA request can be calculated with:

$$t_{COMPx} = t_{RTICK} \times (RTICPUCy + 1) \times RTIUDCPy$$

if  $RTICPUCy \neq 0$ ,

$$t_{COMPx} = t_{RTICK} \times (2^{32} + 1) \times RTIUDCPy$$

if  $RTIUDCPy = 0$ ,

$$t_{COMPx} = t_{RTICK} \times (RTICPUCy + 1) \times 2^{32} \quad (2)$$

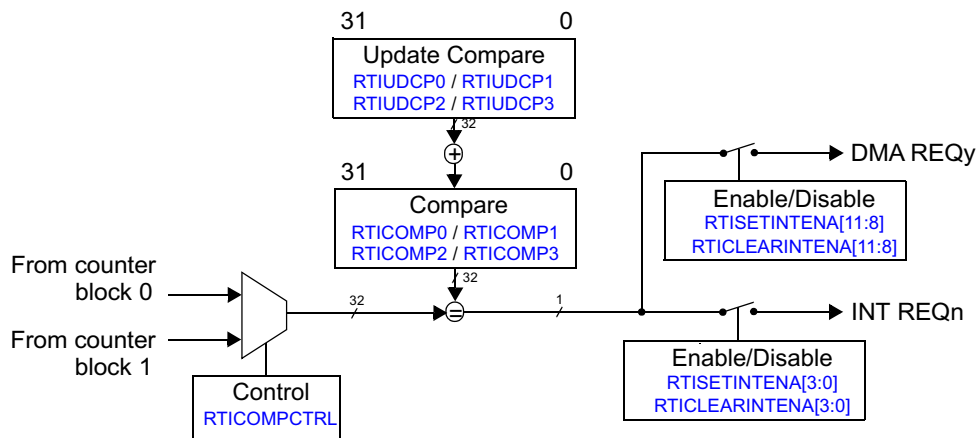


Figure 14-3. Compare Unit Block Diagram (shows only 1 of 4 blocks for simplification)

Another interrupt that can be generated is the overflow interrupt (OVLINTx) in case the RTIFRCx counter overflows.

The interrupts/DMA requests can be enabled in the RTISETINTENA register and disabled in the RTICLEARINTENA register. The RTIINTFLAG register shows the pending interrupts.

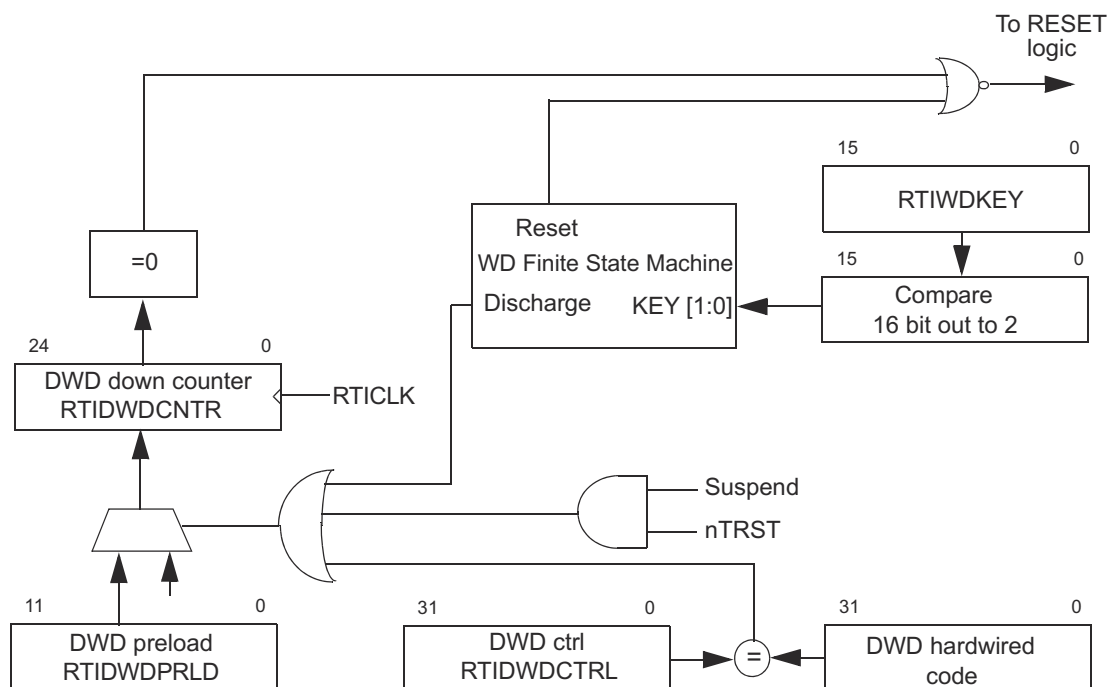
### 14.2.3 RTI Clocking

The counter blocks are clocked with RTICK.

A clock supervision for the NTUX clocking scheme is implemented to avoid missing operating system ticks.

### 14.2.4 Digital Watchdog (DWD)

The digital watchdog (DWD) is an optional safety diagnostic which can detect a runaway CPU and generate either a reset or NMI (non-maskable interrupt) response. It generates resets or NMIs after a programmable period, and generates a reset if no correct key sequence was written to the RTIWDKEY register. Figure 14-4 illustrates the DWD.



**Figure 14-4. Digital Watchdog**

#### 14.2.4.1 Digital Watchdog (DWD)

The DWD is disabled by default. If it should be used, it must be enabled by writing a 32-bit value to the RTIDWDCTRL register.

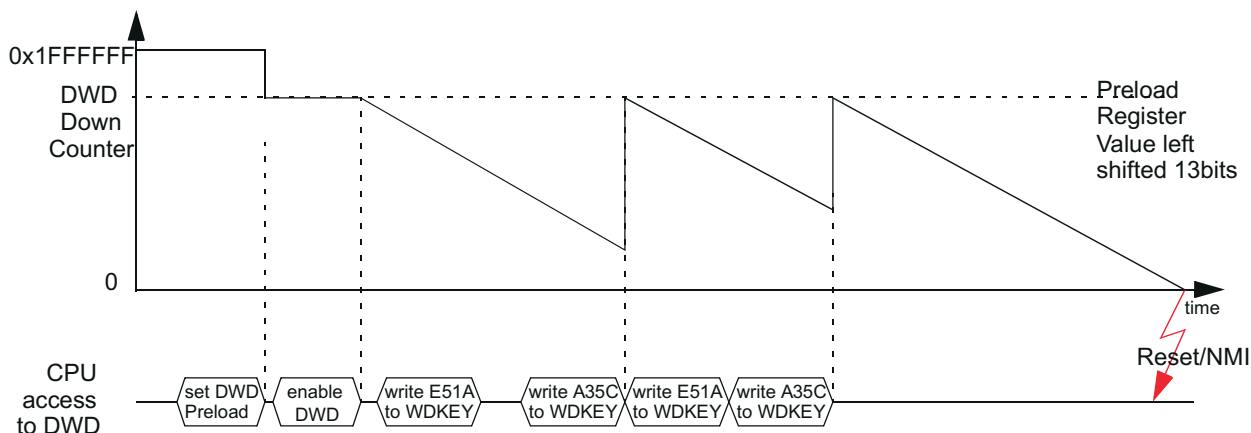
##### Note

Once the DWD is enabled, it cannot be disabled except by system reset or power on reset.

If the correct key sequence is written to the RTIWDKEY register (0xE51A followed by 0xA35C), the 25-bit DWD down counter is reloaded with the left justified 12-bit preload value stored in RTIDWDPRLD. If an incorrect value is written, a watchdog reset will occur immediately. Also a reset or NMI will be generated when the DWD down counter is decremented to 0.

While the device is in suspend mode (halting debug mode), the DWD down counter keeps the value it had when entering suspend mode.

The DWD down counter will be decremented with the RTICLK frequency.



**Figure 14-5. DWD Operation**

The expiration time of the DWD down counter can be determined with the following equation:

$$t_{exp} = (DWDPRLD + 1) \times 2^{13}/RTICLK$$

where

$$DWDPRLD = 0 \dots 4095$$

##### Note

Care should be taken to ensure that the CPU write to the watchdog register is made allowing time for the write to propagate to the RTI.

#### 14.2.4.2 Digital Windowed Watchdog (DWWD)

In addition to the time-out boundary configurable via the digital watchdog discussed in [Section 14.2.4.1](#), for enhanced safety metrics it is desirable to check for a watchdog "pet" within a time window rather than using a single time threshold. This is enabled by the digital windowed watchdog (DWWD) feature.

##### • Functional Behavior

The DWWD opens a configurable time window in which the watchdog must be serviced. Any attempt to service the watchdog outside this time window, or a failure to service the watchdog in this time window, will cause

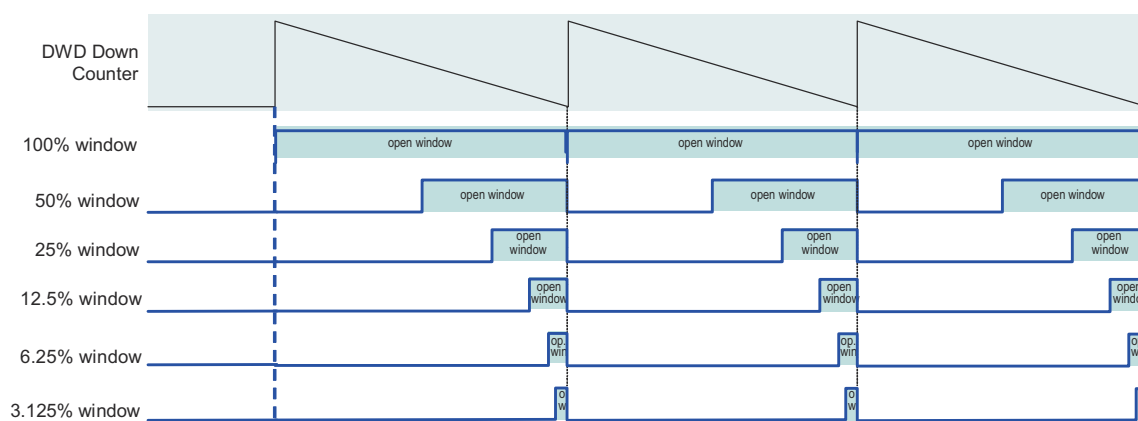
the watchdog to generate either a reset to the CPU. This is controlled by configuring the RTIWWDRXNCTRL register. As with the DWD, the DWWD is disabled after power on reset. When the DWWD is configured to generate a non-maskable interrupt on a window violation, the watchdog counter continues to count down. The NMI handler needs to clear the watchdog violation status flag(s) and then service the watchdog by writing the correct sequence in the watchdog key register. This service will cause the watchdog counter to get reloaded from the preload value and start counting down. If the NMI handler does not service the watchdog in time, it could count down all the way to zero and wrap around. If the NMI Handler does not service the watchdog in time, the NMI gets generated continuously, each time the counter counts to '0'.

The DWWD uses the Digital Watchdog (DWD) preload register (RTIDWDPRLD) setting to define the end-time of the window. The start-time of the window is defined by a window size configuration register(RTIWWDSIZECTRL).

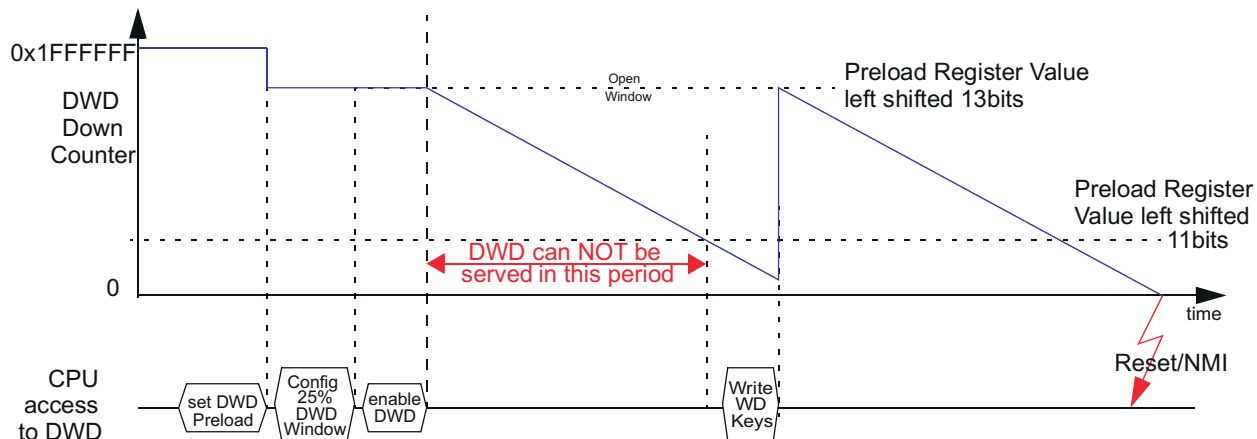
The default window size is set to 100%, which corresponds to the DWD functionality of a time-out-only watchdog. The window size can be selected (through register RTIWWDSIZECTRL) from among 100%, 50%, 25%, 12.5%, 6.25% and 3.125% as shown in Figure 14-6. The window with the respective size will be opened before the end of the DWD expiration. The user has to serve the watchdog in the window. Otherwise, a reset or NMI will generate. Figure 14-7 shows an DWWD operation example (25% window).

- Configuration of DWWD

The DWWD preload value (same as DWD preload) can only be configured when the DWWD counter is disabled. The window size and watchdog reaction to a violation can be configured even after the watchdog has been enabled. Any changes to the window size and watchdog reaction configurations will only take effect after the next servicing of the DWWD. This feature can be utilized to dynamically set windows of different sizes based on task execution time, adding a program sequence element to the diagnostic which can improve fault coverage.



**Figure 14-6. Digital Windowed Watchdog Timing Example**



**Figure 14-7. Digital Windowed Watchdog Operation Example (25% Window)**

#### 14.2.5 Halting Debug Mode Behaviour

Once the system enters halting debug mode, the behavior of the RTI depends on the COS (continue on suspend) bit. If the bit is cleared and halting debug mode is active, all counters will stop operation. If the bit is set to one, all counters will be clocked normally and the RTI will work like in normal mode. However, if the external timebase (NTU) is used and the system is in halting debug mode, the timebase control circuit will switch to internal timebase once it detects the missing NTU signal of the suspended communication controller. This will be signaled with an TBINT interrupt so that software can resynchronize after the device exits halting debug mode.

## 14.3 MSS\_RTI Registers

Table 14-1 lists the memory-mapped registers for the MSS\_RTI registers. All register offset addresses not listed in Table 14-1 should be considered as reserved locations and the register contents should not be modified.

**Table 14-1. MSS\_RTI Registers**

Offset	Acronym	Register Name	Section
0h	RTIGCTRL	RTIGCTRL	<a href="#">Go</a>
4h	RTITBCTRL	RTITBCTRL	<a href="#">Go</a>
8h	RTICAPCTRL	RTICAPCTRL	<a href="#">Go</a>
Ch	RTICOMPCTRL	RTICOMPCTRL	<a href="#">Go</a>
10h	RTIFRC0	RTIFRC0	<a href="#">Go</a>
14h	RTIUC0	RTIUC0	<a href="#">Go</a>
18h	RTICPUC0	RTICPUC0	<a href="#">Go</a>
20h	RTICAFRC0	RTICAFRC0	<a href="#">Go</a>
24h	RTICAUC0	RTICAUC0	<a href="#">Go</a>
30h	RTIFRC1	RTIFRC1	<a href="#">Go</a>
34h	RTIUC1	RTIUC1	<a href="#">Go</a>
38h	RTICPUC1	RTICPUC1	<a href="#">Go</a>
40h	RTICAFRC1	RTICAFRC1	<a href="#">Go</a>
44h	RTICAUC1	RTICAUC1	<a href="#">Go</a>
50h	RTICOMP0	RTICOMP0	<a href="#">Go</a>
54h	RTIUDCP0	RTIUDCP0	<a href="#">Go</a>
58h	RTICOMP1	RTICOMP1	<a href="#">Go</a>
5Ch	RTIUDCP1	RTIUDCP1	<a href="#">Go</a>
60h	RTICOMP2	RTICOMP2	<a href="#">Go</a>
64h	RTIUDCP2	RTIUDCP2	<a href="#">Go</a>
68h	RTICOMP3	RTICOMP3	<a href="#">Go</a>
6Ch	RTIUDCP3	RTIUDCP3	<a href="#">Go</a>
70h	RTITBLCOMP	RTITBLCOMP	<a href="#">Go</a>
74h	RTITBHCOMP	RTITBHCOMP	<a href="#">Go</a>
80h	RTISETINT	RTISETINT	<a href="#">Go</a>
84h	RTICLEARINT	RTICLEARINT	<a href="#">Go</a>
88h	RTIINTFLAG	RTIINTFLAG	<a href="#">Go</a>
90h	RTIDWDCTRL	RTIDWDCTRL	<a href="#">Go</a>
94h	RTIDWDPRLD	RTIDWDPRLD	<a href="#">Go</a>
98h	RTIWDSTATUS	RTIWDSTATUS	<a href="#">Go</a>
9Ch	RTIWDKEY	RTIWDKEY	<a href="#">Go</a>
A0h	RTIDWDCNTR	RTIDWDCNTR	<a href="#">Go</a>
A4h	RTIWWDRXNCTRL	RTIWWDRXNCTRL	<a href="#">Go</a>
A8h	RTIWWDSIZECTRL	RTIWWDSIZECTRL	<a href="#">Go</a>
ACH	RTIINTCLRENABLE	RTIINTCLRENABLE	<a href="#">Go</a>
B0h	RTICOMP0CLR	RTICOMP0CLR	<a href="#">Go</a>
B4h	RTICOMP1CLR	RTICOMP1CLR	<a href="#">Go</a>
B8h	RTICOMP2CLR	RTICOMP2CLR	<a href="#">Go</a>
BCh	RTICOMP3CLR	RTICOMP3CLR	<a href="#">Go</a>



Complex bit access types are encoded to fit into small table cells. [Table 14-2](#) shows the codes that are used for access types in this section.

**Table 14-2. MSS\_RTI Access Type Codes**

Access Type	Code	Description
<b>Read Type</b>		
R	R	Read
<b>Write Type</b>		
W	W	Write
<b>Reset or Default Value</b>		
-n		Value after reset or the default value

### 14.3.1 RTIGCTRL Register (Offset = 0h) [Reset = 00000000h]

RTIGCTRL is shown in [Table 14-3](#).

Return to the [Summary Table](#).

Global Control Register starts / stops the counters

**Table 14-3. RTIGCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
19-16	NTUSEL	R/W	0h	NTUSEL: Select NTU signal. These bits determine which NTU input signal is used as external timebase. There are up to four inputs supported with four valid selection combinations. Any invalid selection value written to the NTUSEL bit-field will result in a TIED LOW being used as the NTU signal. The NTU signal will also be TIED LOW in case of a single-bit flip as it will result in an invalid combination of NTUSEL. User and privilege mode (read): 0000 = NTU0 0101 = NTU1 1010 = NTU2 1111 = NTU3 other = tied to '0' Privilege mode (write): 0000 = NTU0 0101 = NTU1 1010 = NTU2 1111 = NTU3 other = tied to '0'
15	COS	R/W	0h	COS: Continue On Suspend. This bit determines if both counters are stopped when the device goes into debug mode or if they continue counting. User and privilege mode (read): 0 = counters are stopped while in debug mode 1 = counters are running while in debug mode Privilege mode (write): 0 = stop counters in debug mode 1 = continue counting in debug mode
14-2	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
1	CNT1EN	R/W	0h	CNT1EN: Counter 1 Enable. The CNT1EN bit starts and stops the operation of counter block 1 (UC1 and FRC1). User and privilege mode (read): 0 = counters are stopped 1 = counters are running Privilege mode (write): 0 = stop counters 1 = start counters Gives the absolute 32 bit destination address (physical).
0	CNT0EN	R/W	0h	CNT0EN: Counter 0 Enable. The CNT0EN bit starts and stops the operation of counter block 0 (UC0 and FRC0). User and privilege mode (read): 0 = counters are stopped 1 = counters are running Privilege mode (write): 0 = stop counters 1 = start counters Gives the absolute 32 bits source address (physical).

### 14.3.2 RTITBCTRL Register (Offset = 4h) [Reset = 00000000h]

RTITBCTRL is shown in [Table 14-4](#).

Return to the [Summary Table](#).

Timebase Control selection which source triggers free running counter 0

**Table 14-4. RTITBCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Reserved
1	INC	R/W	0h	<p>INC: Increment Free Running Counter 0.</p> <p>This bit determines whether the Free Running Counter 0 is automatically incremented if a failing clock on the NTUx signal is detected.</p> <p>User and privilege mode (read):</p> <p>0 = FRC0 will not be incremented</p> <p>1 = FRC0 will be incremented</p> <p>Privilege mode (write):</p> <p>0 = Do not increment FRC0 on failing external clock</p> <p>1 = Increment FRC0 on failing external clock</p>
0	TBEXT	R/W	0h	<p>TBEXT: Timebase External.</p> <p>The Timebase External bit selects whether the Free Running Counter 0 is clocked by the internal Up Counter 0 or from the external signal NTUx.</p> <p>Since setting the TBEXT bit to 1 resets Up Counter 0, Free Running Counter 0 will not be incremented in this occurrence.</p> <p>The only source which is able to increment Free Running Counter 0 is NTUx.</p> <p>When the Timebase Supervisor circuit detects a missing clockedge, then the TBEXT bit is reset.</p> <p>The selection if the external signal should be used, can only be done by software.</p> <p>User and privilege mode (read):</p> <p>0 = UC0 clocks FRC0</p> <p>1 = NTUx clocks FRC0</p> <p>Privilege mode (write):</p> <p>0 = MUX is switched to internal UC0 clocking scheme</p> <p>1 = MUX is switched to external NTUx clocking scheme</p>

### 14.3.3 RTICAPCTRL Register (Offset = 8h) [Reset = 00000000h]

RTICAPCTRL is shown in [Table 14-5](#).

Return to the [Summary Table](#).

Capture Control controls the capture source for the counters

**Table 14-5. RTICAPCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
1	CAPCNTR1	R/W	0h	CAPCNTR 1: Capture Counter 1. This bit determines, which external interrupt source triggers a capture event of both UC1 and FRC1. User and privilege mode (read): 0 = capture event is triggered by Capture Event Source 0 1 = capture event is triggered by Capture Event Source 1 Privilege mode (write): 0 = enable capture event triggered by Capture Event Source 0 1 = enable capture event triggered by Capture Event Source 1
0	CAPCNTR0	R/W	0h	CAPCNTR 0: Capture Counter 0. This bit determines, which external interrupt source triggers a capture event of both UC0 and FRC0. User and privilege mode (read): 0 = capture event is triggered by Capture Event Source 0 1 = capture event is triggered by Capture Event Source 1 Privilege mode (write): 0 = enable capture event triggered by Capture Event Source 0 1 = enable capture event triggered by Capture Event Source 1 11 indexed 10 reserved 01 post-increment 00 constant

#### 14.3.4 RTICOMPCTRL Register (Offset = Ch) [Reset = 00000000h]

RTICOMPCTRL is shown in [Table 14-6](#).

Return to the [Summary Table](#).

Compare Control controls the source for the compare registers

**Table 14-6. RTICOMPCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
12	COMP3SEL	R/W	0h	COMPSEL 3: Compare Select 3. This bit determines the counter with which the compare value hold in compare register 3 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1
11-9	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
8	COMP2SEL	R/W	0h	COMPSEL 2: Compare Select 2. This bit determines the counter with which the compare value hold in compare register 2 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1
7-5	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
4	COMP1SEL	R/W	0h	COMPSEL 1: Compare Select 1. This bit determines the counter with which the compare value hold in compare register 1 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1
3-1	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
0	COMP0SEL	R/W	0h	COMPSEL 0: Compare Select 0. This bit determines the counter with which the compare value hold in compare register 0 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1

### 14.3.5 RTIFRC0 Register (Offset = 10h) [Reset = 00000000h]

RTIFRC0 is shown in [Table 14-7](#).

Return to the [Summary Table](#).

Free Running Counter 0 current value of free running counter 0

**Table 14-7. RTIFRC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FRC0	R/W	0h	<p>FRC0: Free Running Counter 0.</p> <p>This registers holds the current value of the Free Running Counter 0 and will be updated continuously.</p> <p>User and privilege mode (read): current value of the counter</p> <p>Privilege mode (write): The counter can be preset by writing to this register.</p> <p>The counter increments then from this written value upwards.</p> <p>Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0.</p>

### 14.3.6 RTIUC0 Register (Offset = 14h) [Reset = 00000000h]

RTIUC0 is shown in [Table 14-8](#).

Return to the [Summary Table](#).

Up Counter 0 current value of prescale counter 0

**Table 14-8. RTIUC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	UC0	R/W	0h	<p>UC0: Up Counter 0.</p> <p>This registers holds the current value of the Up Counter 0 and prescales the RTI clock.</p> <p>It will be only updated by a previous read of Free Running Counter 0.</p> <p>This gives effectively a 64 bit read of both counters, without having the problem of a counter being updated between two consecutive reads on Up Counter 0 and Free Running Counter 0.</p> <p>User and privilege mode (read): value of the counter when the Free Running Counter 0 was read Privilege mode (write): the counter can be preset by writing to this register.</p> <p>The counter increments then from this written value upwards.</p> <p>Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0.</p> <p>Note: Preset value concern If the preset value is bigger than the compare value stored in register RTICPUC0 then it can take a long time until a compare matches, since RTIUC0 has to count up until it overflows.</p>

### 14.3.7 RTICPUC0 Register (Offset = 18h) [Reset = 00000000h]

RTICPUC0 is shown in [Table 14-9](#).

Return to the [Summary Table](#).

Compare Up Counter 0 compare value compared with prescale counter 0

**Table 14-9. RTICPUC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPUC0	R/W	0h	<p>This registers holds the compare value, which is compared with the Up Counter 0.</p> <p>When the compare matches, Free Running counter 0 is incremented.</p> <p>The Up Counter is set to zero when the counter value matches the CPUC0 value.</p> <p>The value set in this prescales the RTI clock.</p> <p>If CPUC</p> <p>0 = 0: then, frequency = <math>RTICLK / (2^{32})</math> If CPUC0 <math>\neq</math> 0: then , frequency = <math>RTICLK / (CPUC0 + 1)</math></p> <p>User and privilege mode (read): current compare value</p> <p>Privilege mode (write when TBEXT = 0): the compare value is updated</p> <p>Privilege mode (write when TBEXT = 1): the compare value is not changed</p>



### 14.3.8 RTICAFRC0 Register (Offset = 20h) [Reset = 00000000h]

RTICAFRC0 is shown in [Table 14-10](#).

Return to the [Summary Table](#).

Capture Free Running Counter 0 current value of free running counter 0 on external event

**Table 14-10. RTICAFRC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CAFR0	R/W	0h	CAFR0: Capture Free Running Counter 0. This registers captures the current value of the Free Running Counter 0 when a event occurs, controlled by the external capture control block. User and privilege mode (read): value of Free Running Counter 0 on a capture event

### 14.3.9 RTICAUC0 Register (Offset = 24h) [Reset = 00000000h]

RTICAUC0 is shown in [Table 14-11](#).

Return to the [Summary Table](#).

Capture Up Counter 0 current value of prescale counter 0 on external event

**Table 14-11. RTICAUC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CAUC0	R/W	0h	<p>CAUC0: Capture Up Counter 0.</p> <p>This registers captures the current value of the Up Counter 0 when a event occurs, controlled by the external capture control block. The read sequence has to be the same as with Up Counter 0 and Free Running Counter 0.</p> <p>So the RTICAFRC0 register has to be read first, before the RTICAUC0 register is read.</p> <p>This sequence ensures that the value of the RTICAUC0 register is the corresponding value to the RTICAFRC0 register, even if another capture event happens in between the two reads.</p> <p>User and privilege mode (read): value of Up Counter 0 on a capture event</p>

### 14.3.10 RTIFRC1 Register (Offset = 30h) [Reset = 00000000h]

RTIFRC1 is shown in [Table 14-12](#).

Return to the [Summary Table](#).

Free Running Counter 1 current value of free running counter 1

**Table 14-12. RTIFRC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	FRC1	R/W	0h	<p>FRC1: Free Running Counter 1.</p> <p>This registers holds the current value of the Free Running Counter 1 and will be updated continuously.</p> <p>User and privilege mode (read): current value of the counter</p> <p>Privilege mode (write): The counter can be preset by writing to this register.</p> <p>The counter increments then from this written value upwards.</p> <p>Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC1 and RTIFRC1.</p>

### 14.3.11 RTIUC1 Register (Offset = 34h) [Reset = 00000000h]

RTIUC1 is shown in [Table 14-13](#).

Return to the [Summary Table](#).

Up Counter 1 current value of prescale counter 1

**Table 14-13. RTIUC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	UC1	R/W	0h	<p>UC1: Up Counter 1.</p> <p>This registers holds the current value of the Up Counter 1 and prescales the RTI clock.</p> <p>It will be only updated by a previous read of Free Running Counter 1. This gives effectively a 64 bit read of both counters, without having the problem of a counter being updated between two consecutive reads on Up Counter 1 and Free Running Counter 1.</p> <p>User and privilege mode (read): value of the counter when the Free Running Counter 1 was read Privilege mode (write): the counter can be preset by writing to this register.</p> <p>The counter increments then from this written value upwards.</p> <p>Note: Presetting counters If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC1 and RTIFRC1.</p> <p>Note: Preset value concern If the preset value is bigger than the compare value stored in register RTICPUC1 then it can take a long time until a compare matches, since RTIUC1 has to count up until it overflows.</p>

### 14.3.12 RTICPUC1 Register (Offset = 38h) [Reset = 00000000h]

RTICPUC1 is shown in [Table 14-14](#).

Return to the [Summary Table](#).

Compare Up Counter 1 compare value compared with prescale counter 1

**Table 14-14. RTICPUC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CPUC1	R/W	0h	<p>This registers holds the compare value, which is compared with the Up Counter 1.</p> <p>When the compare matches, Free Running Counter 1 is incremented.</p> <p>The Up Counter is set to zero when the counter value matches the CPUC1 value.</p> <p>The value set in this prescales the RTI clock.</p> <p>If CPUC</p> <p>1 = 0: then, frequency = RTICLK/ (<math>2^{32}</math>) If CPUC1 <math>\neq</math> 0: then , frequency = RTICLK/(CPUC1 + 1) User and privilege mode (read): current compare value Privilege mode (write when TBEXT = 0): the compare value is updated Privilege mode (write when TBEXT = 1): the compare value is not changed</p>

### 14.3.13 RTICAFRC1 Register (Offset = 40h) [Reset = 00000000h]

RTICAFRC1 is shown in [Table 14-15](#).

Return to the [Summary Table](#).

Capture Free Running Counter 1 current value of free running counter 1 on external event

**Table 14-15. RTICAFRC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CAFRC1	R/W	0h	CAFRC1: Capture Free Running Counter 1. This registers captures the current value of the Free Running Counter 1 when a event occurs, controlled by the external capture control block. User and privilege mode (read): value of Free Running Counter 1 on a capture event

### 14.3.14 RTICAUC1 Register (Offset = 44h) [Reset = 00000000h]

RTICAUC1 is shown in [Table 14-16](#).

Return to the [Summary Table](#).

Capture Up Counter 1 current value of prescale counter 1 on external event

**Table 14-16. RTICAUC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CAUC1	R/W	0h	<p>CAUC1: Capture Up Counter 1.</p> <p>This registers captures the current value of the Up Counter 1 when a event occurs, controlled by the external capture control block. The read sequence has to be the same as with Up Counter 1 and Free Running Counter 1.</p> <p>So the RTICAFRC1 register has to be read first, before the RTICAUC1 register is read.</p> <p>This sequence ensures that the value of the RTICAUC1 register is the corresponding value to the RTICAFRC1 register, even if another capture event happens in between the two reads.</p> <p>User and privilege mode (read): value of Up Counter 1 on a capture event</p>

### 14.3.15 RTICOMP0 Register (Offset = 50h) [Reset = 00000000h]

RTICOMP0 is shown in [Table 14-17](#).

Return to the [Summary Table](#).

Compare 0 compare value to be compared with the counters

**Table 14-17. RTICOMP0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP0	R/W	0h	<p>COMP0: Compare 0.</p> <p>This registers holds a compare value, which is compared with the counter selected in the compare control logic.</p> <p>If the Free Running Counter matches the compare value, an interrupt is flagged.</p> <p>With this register it is also possible to initiate a DMA request.</p> <p>User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.</p>



### 14.3.16 RTIUDCP0 Register (Offset = 54h) [Reset = 00000000h]

RTIUDCP0 is shown in [Table 14-18](#).

Return to the [Summary Table](#).

Update Compare 0 value to be added to the compare register 0 value on compare match

**Table 14-18. RTIUDCP0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	UDCP0	R/W	0h	UDCP0: Update Compare 0 Register. This registers holds a value, which is added to the value in the compare 0 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare 0 register on the next compare match Privilege mode (write): new update value

### 14.3.17 RTICOMP1 Register (Offset = 58h) [Reset = 00000000h]

RTICOMP1 is shown in [Table 14-19](#).

Return to the [Summary Table](#).

Compare 1 compare value to be compared with the counters

**Table 14-19. RTICOMP1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP1	R/W	0h	<p>COMP1: compare1.</p> <p>This registers holds a compare value, which is compared with the counter selected in the compare control logic.</p> <p>If the Free Running Counter matches the compare value, an interrupt is flagged.</p> <p>With this register it is also possible to initiate a DMA request.</p> <p>User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.</p>

### 14.3.18 RTIUDCP1 Register (Offset = 5Ch) [Reset = 00000000h]

RTIUDCP1 is shown in [Table 14-20](#).

Return to the [Summary Table](#).

Update Compare 1 value to be added to the compare register 1 value on compare match

**Table 14-20. RTIUDCP1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	UDCP1	R/W	0h	UDCP1: Update compare1 Register. This registers holds a value, which is added to the value in the compare1 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare1 register on the next compare match Privilege mode (write): new update value

### 14.3.19 RTICOMP2 Register (Offset = 60h) [Reset = 00000000h]

RTICOMP2 is shown in [Table 14-21](#).

Return to the [Summary Table](#).

Compare 2 compare value to be compared with the counters

**Table 14-21. RTICOMP2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP2	R/W	0h	<p>COMP2: compare 2.</p> <p>This registers holds a compare value, which is compared with the counter selected in the compare control logic.</p> <p>If the Free Running Counter matches the compare value, an interrupt is flagged.</p> <p>With this register it is also possible to initiate a DMA request.</p> <p>User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.</p>

### 14.3.20 RTIUDCP2 Register (Offset = 64h) [Reset = 00000000h]

RTIUDCP2 is shown in [Table 14-22](#).

Return to the [Summary Table](#).

Update Compare 2 value to be added to the compare register 2 value on compare match

**Table 14-22. RTIUDCP2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	UDCP2	R/W	0h	UDCP2: Update compare 2 Register. This registers holds a value, which is added to the value in the compare 2 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare 2 register on the next compare match Privilege mode (write): new update value

### 14.3.21 RTICOMP3 Register (Offset = 68h) [Reset = 00000000h]

RTICOMP3 is shown in [Table 14-23](#).

Return to the [Summary Table](#).

Compare 3 compare value to be compared with the counters

**Table 14-23. RTICOMP3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP3	R/W	0h	<p>COMP3: compare 3.</p> <p>This registers holds a compare value, which is compared with the counter selected in the compare control logic.</p> <p>If the Free Running Counter matches the compare value, an interrupt is flagged.</p> <p>With this register it is also possible to initiate a DMA request.</p> <p>User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.</p>

### 14.3.22 RTIUDCP3 Register (Offset = 6Ch) [Reset = 00000000h]

RTIUDCP3 is shown in [Table 14-24](#).

Return to the [Summary Table](#).

Update Compare 3 value to be added to the compare register 3 value on compare match

**Table 14-24. RTIUDCP3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	UDCP3	R/W	0h	UDCP3: Update compare 3 Register. This registers holds a value, which is added to the value in the compare 3 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention. User and privilege mode (read): value to be added to the compare 3 register on the next compare match Privilege mode (write): new update value

### 14.3.23 RTITBLCOMP Register (Offset = 70h) [Reset = 00000000h]

RTITBLCOMP is shown in [Table 14-25](#).

Return to the [Summary Table](#).

Timebase Low Compare compare value to activate edge detection circuit

**Table 14-25. RTITBLCOMP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TBLCOMP	R/W	0h	<p>TBLCOMP: Timebase Low Compare Value.</p> <p>This value determines when the edge detection circuit starts monitoring the NTUx signal.</p> <p>It will be compared with Up Counter 0.</p> <p>User and privilege mode (read): current compare value</p> <p>Privilege mode (write when TBEXT = 0): the compare value is updated</p> <p>Privilege mode (write when TBEXT = 1): the compare value is not changed</p> <p>Note: Reset behavior A reset does not generate a compare match.</p>



### 14.3.24 RTITBHCOMP Register (Offset = 74h) [Reset = 00000000h]

RTITBHCOMP is shown in [Table 14-26](#).

Return to the [Summary Table](#).

Timebase High Compare compare value to deactivate edge detection circuit

**Table 14-26. RTITBHCOMP Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TBHCMP	R/W	0h	<p>TBHCMP: Timebase High Compare Value.</p> <p>This value determines when the edge detection circuit will stop monitoring the NTUx signal.</p> <p>It will be compared with Up Counter 0.</p> <p>RTITBHCOMP has to be less than RTICPUC0, since RTIUC0 will be reset when RTICPUC0 is reached.</p> <p>Example: The NTUx edge detection circuit should be active +/- 10 RTICLK cycles around RTICPUC0.</p> <p>RTICPUC</p> <p>0 = 0x00000050 RTITBLCOMP = 0x000046 RTITBHCOMP = 0x00000009</p> <p>User and privilege mode (read): current compare value</p> <p>Privilege mode (write when TBEXT = 0): the compare value is updated</p> <p>Privilege mode (write when TBEXT = 1): the compare value is not changed</p> <p>Note: Reset behavior A reset does not generate a compare match.</p>

### 14.3.25 RTISETINT Register (Offset = 80h) [Reset = 00000000h]

RTISETINT is shown in [Table 14-27](#).

Return to the [Summary Table](#).

Set Interrupt Enable sets interrupt enable bits in RTIINTCTRL without having to do a read-modify-write operation

**Table 14-27. RTISETINT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
18	SETOVL1INT	R/W	0h	SETOVL1INT: Set Free Running Counter 1 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
17	SETOVL0INT	R/W	0h	SETOVL0INT: Set Free Running Counter 0 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
16	SETTBINT	R/W	0h	SETTBINT: Set Timebase Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
15-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11	SETDMA3	R/W	0h	SETDMA 3: Set Compare DMA Request 3. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
10	SETDMA2	R/W	0h	SETDMA 2: Set Compare DMA Request 2. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
9	SETDMA1	R/W	0h	SETDMA 1: Set Compare DMA Request 1. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
8	SETDMA0	R/W	0h	SETDMA 0: Set Compare DMA Request 0. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
7-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect

**Table 14-27. RTISETINT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
3	SETINT3	R/W	0h	SETINT 3: Set Compare Interrupt 3. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged
2	SETINT2	R/W	0h	SETINT 2: Set Compare Interrupt 2. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
1	SETINT1	R/W	0h	SETINT 1: Set Compare Interrupt 1. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
0	SETINT0	R/W	0h	SETINT 0: Set Compare Interrupt 0. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt

### 14.3.26 RTICLEARINT Register (Offset = 84h) [Reset = 00000000h]

RTICLEARINT is shown in [Table 14-28](#).

Return to the [Summary Table](#).

Clear Interrupt Enable clears interrupt enable bits in RTIINTCTRL without having to do a read-modify-write operation

**Table 14-28. RTICLEARINT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
18	CLEAROVL1INT	R/W	0h	CLEAROVL1INT: CLEAR Free Running Counter 1 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
17	CLEAROVL0INT	R/W	0h	CLEAROVL0INT: CLEAR Free Running Counter 0 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
16	CLEARTBINT	R/W	0h	CLEARTBINT: CLEAR Timebase Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
15-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11	CLEARDMA3	R/W	0h	CLEARDMA 3: CLEAR Compare DMA Request 3. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
10	CLEARDMA2	R/W	0h	CLEARDMA 2: CLEAR Compare DMA Request 2. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
9	CLEARDMA1	R/W	0h	CLEARDMA 1: CLEAR Compare DMA Request 1. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
8	CLEARDMA0	R/W	0h	CLEARDMA 0: CLEAR Compare DMA Request 0. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request

**Table 14-28. RTICLEARINT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3	CLEARINT3	R/W	0h	CLEARINT 3: CLEAR Compare Interrupt 3. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
2	CLEARINT2	R/W	0h	CLEARINT 2: CLEAR Compare Interrupt 2. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
1	CLEARINT1	R/W	0h	CLEARINT 1: CLEAR Compare Interrupt 1. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
0	CLEARINT0	R/W	0h	CLEARINT 0: CLEAR Compare Interrupt 0. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

### 14.3.27 RTIINTFLAG Register (Offset = 88h) [Reset = 00000000h]

RTIINTFLAG is shown in [Table 14-29](#).

Return to the [Summary Table](#).

Interrupt Flags interrupt pending bits

**Table 14-29. RTIINTFLAG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
18	OVL1INT	R/W	0h	OVL1INT: Free Running Counter 1 Overflow Interrupt Flag. User and privilege mode (read): determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
17	OVL0INT	R/W	0h	OVL0INT: Free Running Counter 0 Overflow Interrupt Flag. User and privilege mode (read): determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
16	TBINT	R/W	0h	User and privilege mode (read): this flag is set when the TBEXT bit is cleared by detection of a missing external clockedge. It will not be set by clearing TBEXT by software. determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
15-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3	INT3	R/W	0h	INT 3: Interrupt Flag 3. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
2	INT2	R/W	0h	INT 2: Interrupt Flag 2. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
1	INT1	R/W	0h	INT 1: Interrupt Flag 1. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0
0	INT0	R/W	0h	INT 0: Interrupt Flag 0. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0

### 14.3.28 RTIDWDCTRL Register (Offset = 90h) [Reset = 00000000h]

RTIDWDCTRL is shown in [Table 14-30](#).

Return to the [Summary Table](#).

Digital Watchdog Control Enables the Digital Watchdog

**Table 14-30. RTIDWDCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	DWDCTRL	R/W	0h	<p>DWDCTRL: Digital Watchdog Control.</p> <p>User and privileged mode (read): 0x5312ACED = DWD counter is disabled.</p> <p>This is the default value.</p> <p>0xA98559DA = DWD counter is enabled Any other value = DWD counter state is unchanged (enabled or disabled) Privileged mode (write): 0xA98559DA = DWD counter is enabled Any other value = State of DWD counter is unchanged (stays enabled or disabled) Note: One-Write Functionality of DWDCTRL Register The RTIDWDCTRL register implements a one-write functionality, such that the application cannot write to this register more than once.</p> <p>Writing the default value will not enable the watchdog as described above.</p> <p>Writing the enable value will start the watchdog counters.</p> <p>A write to RTIDWDCTRL will only be enabled after a system reset again.</p>

### 14.3.29 RTIDWDPRLD Register (Offset = 94h) [Reset = 00000000h]

RTIDWDPRLD is shown in [Table 14-31](#).

Return to the [Summary Table](#).

Digital Watchdog Preload sets the expiration time of the Digital Watchdog

**Table 14-31. RTIDWDPRLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11-0	DWDPRLD	R/W	0h	DWDPRLD: Digital Watchdog Preload Value. User and privilege mode (read): A read from this register in any CPU mode returns the current preload value. Privilege mode (write): If the DWD is always enabled after reset is released: The DWD starts counting down from the reset value of the counter, that is, 0x002DFFFF. The application can configure the DWD preload register any time before this down counter expires. When the application services the DWD, the preload register contents are copied left-justified into the DWD down counter and it starts counting down from that value. If the DWD is implemented such that the down counter is enabled by software: The DWD preload register can be configured only when the DWD is disabled. Therefore, the application can only configure the DWD preload register before it enables the DWD down counter. The expiration time of the DWD Down Counter can be determined with following equation: $t_{exp} = (RTIDWDPRLD + 1) \times 2^{13} / RTICLK1$ where: RTIDWDPRLD = 0...4095



### 14.3.30 RTIWDSTATUS Register (Offset = 98h) [Reset = 00000000h]

RTIWDSTATUS is shown in [Table 14-32](#).

Return to the [Summary Table](#).

Watchdog Status reflects the status of Analog and Digital Watchdog

**Table 14-32. RTIWDSTATUS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
5	DWWD_ST	R/W	0h	DWWD ST: Windowed Watchdog Status. This bit denotes whether the time-window defined by the windowed watchdog configuration has been violated, or if a wrong key or key sequence was written to service the watchdog. User and privilege mode (read): 0 = no time-window violation has occurred. 1 = a time-window violation has occurred. The watchdog will generate either a system reset or a non-maskable interrupt to the CPU in this case. Privilege mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0. This will also clear all other status flags in the RTIWDSTATUS register except for the AWD ST flag. Clearing of the status flags will deassert the non-maskable interrupt generated due to violation of the DWWD.
4	ENDTIMEVIOL	R/W	0h	END TIME VIOL: Windowed Watchdog End Time Violation Status. This bit denotes whether the end-time defined by the windowed watchdog configuration has been violated. This bit is effectively a copy of the DWD ST status flag. User and privilege mode (read): 0 = no end-time window violation has occurred. 1 = the end-time defined by the windowed watchdog configuration has been violated. Privilege mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0.
3	STARTTIMEVIOL	R/W	0h	START TIME VIOL: Windowed Watchdog Start Time Violation Status. This bit denotes whether the start-time defined by the windowed watchdog configuration has been violated. This indicates that the WWD was serviced before the service window was opened. User and privilege mode (read): 0 = no start-time window violation has occurred. 1 = the start-time defined by the windowed watchdog configuration has been violated. Privilege mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0.
2	KEYST	R/W	0h	KEYST: Watchdog KeyStatus. This bit denotes a reset generated by a wrong key or a wrong key-sequence written to the RTIWDKEY register. User and privilege mode (read): 0 = no wrong key or key-sequence written 1 = wrong key or key-sequence written to RTIWDKEY register Privilege mode (write): 0 = leaves the current value unchanged 1 = clears the bit to 0

**Table 14-32. RTIWDSTATUS Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	DWDST	R/W	0h	DWDST: Digital Watchdog Status. This bit is effectively a copy of the END TIME VIOL status flag and is maintained for compatibility reasons. User and privilege mode (read): 0 = DWD timeout period not expired 1 = DWD timeout period has expired Privilege mode (write): 0 = leaves the current value unchanged 1 = clears the bit to 0
0	AWDST	R/W	0h	AWDST: Analog Watchdog Status. User and privilege mode (read): 0 = AWD pin 0 → 1 threshold not exceeded 1 = AWD pin 0 → 1 threshold exceeded Privilege mode (write): 0 = leaves the current value unchanged 1 = clears the bit to 0

### 14.3.31 RTIWDKEY Register (Offset = 9Ch) [Reset = 00000000h]

RTIWDKEY is shown in [Table 14-33](#).

Return to the [Summary Table](#).

Watchdog Key correct written key values discharge the external capacitor

**Table 14-33. RTIWDKEY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
15-0	WDKEY	R/W	0h	WDKEY: Watchdog Key. User and privilege mode reads are indeterminate. Privilege mode (write): A write of 0xE51A followed by 0xA35C in two separate write operations defines the Key Sequence and discharges the watchdog capacitor. This also causes the upper 12 bits of the DWD down counter to be reloaded with the contents of the DWD preload register and the lower 13 bits to become all 1's. Writing any other value causes a digital watchdog reset, as shown in <a href="#">Table 1-3</a> . Note: Register write access time precaution The user has to take into account that the write to the register takes 3 VCLK cycle. This needs to be considered for the AWD/DWD expiration calculation.

### 14.3.32 RTIDWDCNTR Register (Offset = A0h) [Reset = 00000000h]

RTIDWDCNTR is shown in [Table 14-34](#).

Return to the [Summary Table](#).

Digital Watchdog Down Counter current value of DWD down counter

**Table 14-34. RTIDWDCNTR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
24-0	DWDCNTR	R/W	0h	DWDCNTR: Digital Watchdog Down Counter. The value of the DWDCNTR after a system reset is 0x002D_FFFF. When the DWD is enabled and the DWD counter starts counting down from this value with an RTICK1 time base of 3MHz, a watchdog reset will be generated in 1 second. User and privilege mode (read): Reads return the current counter value. Privilege mode (write): Writes don't have an effect.

### 14.3.33 RTIWWDRXNCTRL Register (Offset = A4h) [Reset = 00000000h]

RTIWWDRXNCTRL is shown in [Table 14-35](#).

Return to the [Summary Table](#).

Windowed Watchdog Reaction Control configures the windowed watchdog to either generate a non-maskable interrupt to the CPU or to generate a system reset

**Table 14-35. RTIWWDRXNCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3-0	WWDRXN	R/W	0h	WWDRXN: Digital Windowed Watchdog Reaction. User and privilege mode (read), privileged mode (write): 0x5 = This is the default value. The windowed watchdog will cause a reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. 0xA = The windowed watchdog will generate a non-maskable interrupt to the CPU if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. Writing any other value will cause a system reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. Note: Configuration of DWWD Reaction The DWWD reaction can be selected by the application even when the DWWD counter is already enabled. If a change to the WWDRXN is made before the watchdog service window is opened, then the change in the configuration takes effect immediately. If a change to the WWDRXN is made when the watchdog service window is already open, then the change in configuration takes effect only after the watchdog is serviced.

### 14.3.34 RTIWWDSIZECTRL Register (Offset = A8h) [Reset = 00000000h]

RTIWWDSIZECTRL is shown in [Table 14-36](#).

Return to the [Summary Table](#).

Windowed Watchdog Size Control configures the size of the window for the digital windowed watchdog

**Table 14-36. RTIWWDSIZECTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	WWDSIZE	R/W	0h	<p>WWDSIZE: Digital Windowed Watchdog Window Size.</p> <p>User and privilege mode (read), privileged mode (write):</p> <p>Value written to WWDSIZE Window Size 0x00000005 100% (Functionality same as the time-out digital watchdog.) 0x00000050 50% 0x00000500 25% 0x00005000 12.5% 0x00050000 6.25% 0x00500000 3.125% Any other value 3.125% Note: Incorrect value being written to watchdog window size control register If an incorrect value is written to the WWDSIZE field, or if a system disturbance causes the WWDSIZE field to have a value other than 0x5, 0x50, 0x500, 0x5000, 0x50000, or 0x500000, then the window size will be configured to be 3.125%.</p> <p>This increases the chances of getting a reset due to the windowed watchdog, which enables the system to handle the cause for the incorrect configuration.</p> <p>Note: Configuration of DWWD Window Size The DWWD window size can be selected by the application even when the DWWD counter is already enabled.</p> <p>If a change to the WWDSIZE is made before the watchdog service window is opened, then the change in the configuration takes effect immediately.</p> <p>If a change to the WWDSIZE is made when the watchdog service window is already open, then</p>

### 14.3.35 RTIINTCLRENABLE Register (Offset = ACh) [Reset = 00000000h]

RTIINTCLRENABLE is shown in [Table 14-37](#).

Return to the [Summary Table](#).

RTI Compare Interrupt Clear Enable enable the auto clear functionality for each of the compare interrupts

**Table 14-37. RTIINTCLRENABLE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
27-24	INTCLRENABLE3	R/W	0h	INTCLRENABLE3. Enables the auto-clear functionality on the compare 3 interrupt. User and Privileged mode (read): 0x 5 = Auto-clear for compare 3 interrupt is disabled. Any other value = Auto-clear for compare 3 interrupt is enabled. Privileged mode (write): 0x 5 = Disables the auto-clear functionality on the compare 3 interrupt. Any other value = Enables the auto-clear functionality on the compare 3 interrupt.
23-20	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
19-16	INTCLRENABLE2	R/W	0h	INTCLRENABLE2. Enables the auto-clear functionality on the compare 2 interrupt. User and Privileged mode (read): 0x 5 = Auto-clear for compare 2 interrupt is disabled. Any other value = Auto-clear for compare 2 interrupt is enabled. Privileged mode (write): 0x 5 = Disables the auto-clear functionality on the compare 2 interrupt. Any other value = Enables the auto-clear functionality on the compare 2 interrupt.
15-12	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
11-8	INTCLRENABLE1	R/W	0h	INTCLRENABLE1. Enables the auto-clear functionality on the compare 1 interrupt. User and Privileged mode (read): 0x 5 = Auto-clear for compare 1 interrupt is disabled. Any other value = Auto-clear for compare 1 interrupt is enabled. Privileged mode (write): 0x 5 = Disables the auto-clear functionality on the compare 1 interrupt. Any other value = Enables the auto-clear functionality on the compare 1 interrupt.
7-4	RESERVED	R/W	0h	Reserved. Reads return 0 and writes have no effect
3-0	INTCLRENABLE0	R/W	0h	INTCLRENABLE0. Enables the auto-clear functionality on the compare 0 interrupt. User and Privileged mode (read): 0x 5 = Auto-clear for compare 0 interrupt is disabled. Any other value = Auto-clear for compare 0 interrupt is enabled. Privileged mode (write): 0x 5 = Disables the auto-clear functionality on the compare 0 interrupt. Any other value = Enables the auto-clear functionality on the compare 0 interrupt.

### 14.3.36 RTICOMP0CLR Register (Offset = B0h) [Reset = 00000000h]

RTICOMP0CLR is shown in [Table 14-38](#).

Return to the [Summary Table](#).

Compare 0 Clear compare value to be compared with the counter to clear the compare0 interrupt line

**Table 14-38. RTICOMP0CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP0CLR	R/W	0h	COMP0CLR: Compare 0 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the compare 0 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.



### 14.3.37 RTICOMP1CLR Register (Offset = B4h) [Reset = 00000000h]

RTICOMP1CLR is shown in [Table 14-39](#).

Return to the [Summary Table](#).

Compare 1 Clear compare value to be compared with the counter to clear the compare1 interrupt line

**Table 14-39. RTICOMP1CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP1CLR	R/W	0h	COMP1CLR: Compare 1 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the Compare 1 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

### 14.3.38 RTICOMP2CLR Register (Offset = B8h) [Reset = 00000000h]

RTICOMP2CLR is shown in [Table 14-40](#).

Return to the [Summary Table](#).

Compare 2 Clear compare value to be compared with the counter to clear the compare2 interrupt line

**Table 14-40. RTICOMP2CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP2CLR	R/W	0h	COMP2CLR: Compare 2 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the Compare 2 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.

### 14.3.39 RTICOMP3CLR Register (Offset = BCh) [Reset = 00000000h]

RTICOMP3CLR is shown in [Table 14-41](#).

Return to the [Summary Table](#).

Compare 3 Clear compare value to be compared with the counter to clear the compare3 interrupt line

**Table 14-41. RTICOMP3CLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	COMP3CLR	R/W	0h	COMP3CLR: Compare 3 Clear. This registers holds a compare value, which is compared with the counter selected in the compare control logic. If the Free Running Counter matches the compare value, the Compare 3 interrupt or DMA request line is cleared. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Note: Reset behavior A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.



This chapter contains the description of the serial communication interface (SCI) module.

## 15.1 Introduction

The SCI module is a universal asynchronous receiver-transmitter that implements the standard nonreturn to zero format. The SCI can be used to communicate, for example, through an RS-232 port or over a K-line.

### 15.1.1 SCI Features

The following are the features of the SCI module:

- Standard universal asynchronous receiver-transmitter (UART) communication
- Supports full- or half-duplex operation
- Standard nonreturn to zero (NRZ) format
- Double-buffered receive and transmit functions
- Supports two individually enabled interrupt lines: level 0 and level 1
- Configurable frame format of 3 to 13 bits per character based on the following:
  - Data word length programmable from one to eight bits
  - Additional address bit in address-bit mode
  - Parity programmable for zero or one parity bit, odd or even parity
  - Stop programmable for one or two stop bits
- Asynchronous communication mode with no CLK pin
- Two multiprocessor communication formats allow communication between more than two devices
- Sleep mode is available to free CPU resources during multiprocessor communication and then wake up to receive an incoming message
- The 24-bit programmable baud rate supports  $2^{24}$  different baud rates provide high accuracy baud rate selection
- Capability to use Direct Memory Access (DMA) for transmit and receive data
- Four error flags and Five status flags provide detailed information regarding SCI events
- Two external pins: SCIRX and SCITX

---

#### Note

SCI module does not support UART Hardware Flow Control. This feature can be implemented in Software using a General Purpose I/O pin.

---

### 15.1.2 Block Diagram

Three Major components of the SCI Module are:

- Transmitter
- Baud Clock Generator
- Receiver

**Transmitter (TX)** contains two major registers to perform double buffering:

- The transmitter data buffer register (SCITD) contains data loaded by the CPU to be transferred to the shift register for transmission.

- The transmitter shift register (SCITXSHF) loads data from the data buffer (SCITD) and shifts data onto the SCITX pin, one bit at a time.

**Baud Clock Generator**

- A programmable baud generator produces a baud clock scaled from VCLK.

**Receiver (RX)** contains two major registers to perform double buffering:

- The receiver shift register (SCIRXSHF) shifts data in from the SCIRX pin one bit at a time and transfers completed data into the receive data buffer.
- The receiver data buffer register (SCIRD) contains received data transferred from the receiver shift register

The SCI receiver and transmitter are double-buffered, and each has its own separate enable and interrupt bits. The receiver and transmitter can each be operated independently or simultaneously in full duplex mode.

To ensure data integrity, the SCI checks the data it receives for breaks, parity, overrun, and framing errors. The bit rate (baud) is programmable to over 16 million different rates through a 24-bit baud-select register. [Figure 15-1](#) shows the detailed SCI block diagram.

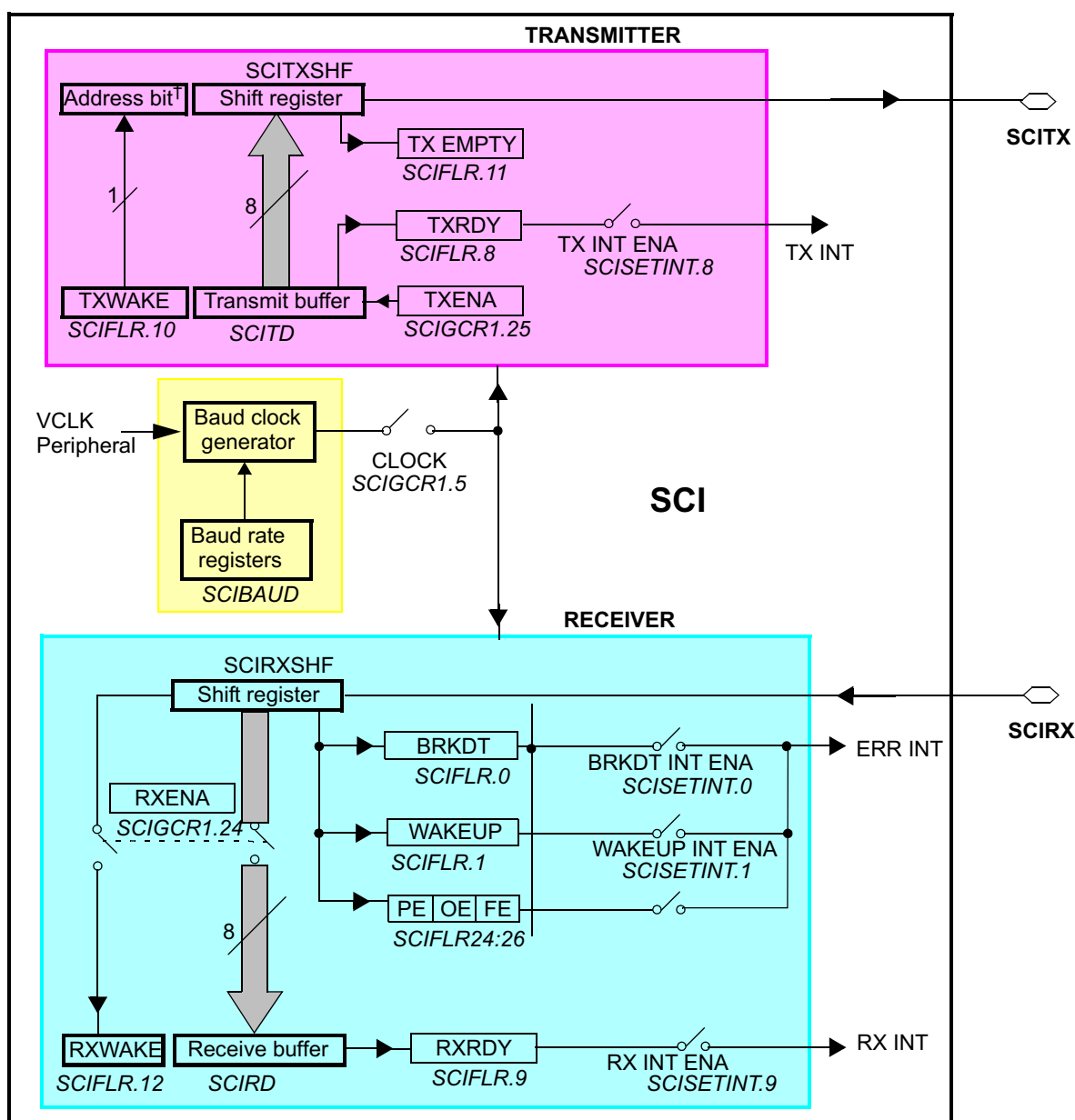


Figure 15-1. Detailed SCI Block Diagram

## 15.2 SCI Communication Formats

The SCI module can be configured to meet the requirements of many applications. Because communication formats vary depending on the specific application, many attributes of the SCI are user configurable. The list below describes these configuration options:

- SCI Frame format
- SCI Timing modes
- SCI Baud rate
- SCI Multiprocessor modes

### 15.2.1 SCI Frame Formats

The SCI uses a programmable frame format. All frames consist of the following:

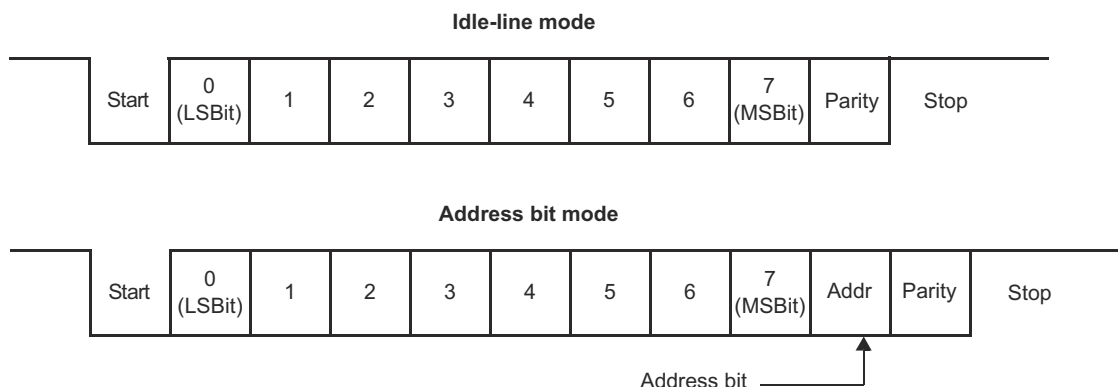
- One start bit
- One to eight data bits
- Zero or one address bit
- Zero or one parity bit
- One or two stop bits

The frame format for both the transmitter and receiver is programmable through the bits in the SCIGCR1 register. Both receive and transmit data is in nonreturn to zero (NRZ) format, which means that the transmit and receive lines are at logic high when idle. Each frame transmission begins with a start bit, in which the transmitter pulls the SCI line low (logic low). Following the start bit, the frame data is sent and received least significant bit first (LSB).

An address bit is present in each frame if the SCI is configured to be in address-bit mode but is not present in any frame if the SCI is configured for idle-line mode. The format of frames with and without the address bit is illustrated in [Figure 15-2](#).

A parity bit is present in every frame when the PARITY ENA bit is set. The value of the parity bit depends on the number of one bits in the frame and whether odd or even parity has been selected via the PARITY ENA bit. Both examples in [Figure 15-2](#) have parity enabled.

All frames include one stop bit, which is always a high level. This high level at the end of each frame is used to indicate the end of a frame to ensure synchronization between communicating devices. Two stop bits are transmitted if the STOP bit in SCIGCR1 register is set. The examples shown in [Figure 15-2](#) use one stop bit per frame.



**Figure 15-2. Typical SCI Data Frame Formats**

### 15.2.2 SCI Timing Mode

The SCI can be configured to use asynchronous or isosynchronous timing using TIMING MODE bit in SCIGCR1 register.

### 15.2.2.1 Asynchronous Timing Mode

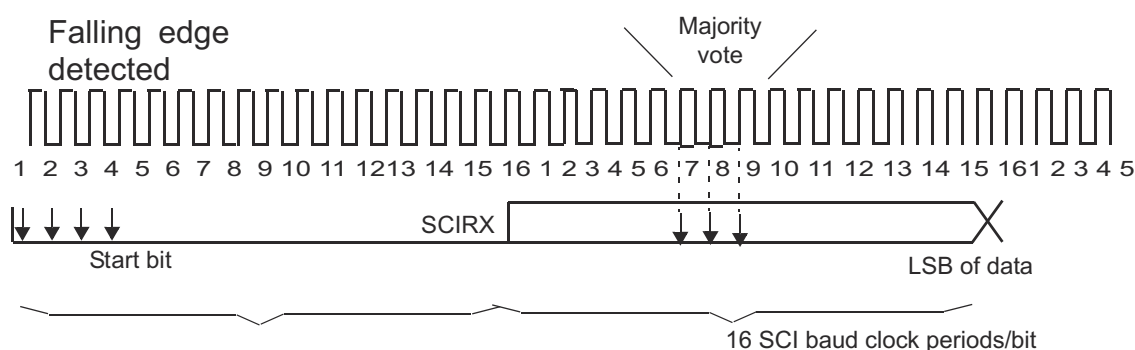
The asynchronous timing mode uses only the receive and transmit data lines to interface with devices using the standard universal asynchronous receiver-transmitter (UART) protocol.

In the asynchronous timing mode, each bit in a frame has a duration of 16 SCI baud clock periods. Each bit therefore consists of 16 samples (one for each clock period). When the SCI is using asynchronous mode, the baud rates of all communicating devices must match as closely as possible. Receive errors result from devices communicating at different baud rates.

With the receiver in the asynchronous timing mode, the SCI detects a valid start bit if the first four samples after a falling edge on the SCIRX pin are of logic level 0. As soon as a falling edge is detected on SCIRX, the SCI assumes that a frame is being received and synchronizes itself to the bus.

To prevent interpreting noise as Start bit SCI expects SCIRX line to be low for at least four contiguous SCI baud clock periods to detect a valid start bit. The bus is considered idle if this condition is not met. When a valid start bit is detected, the SCI determines the value of each bit by sampling the SCIRX line value during the seventh, eighth, and ninth SCI baud clock periods. A majority vote of these three samples is used to determine the value stored in the SCI receiver shift register. By sampling in the middle of the bit, the SCI reduces errors caused by propagation delays and rise and fall times and data line noises. Figure 15-3 illustrates how the receiver samples a start bit and a data bit in asynchronous timing mode.

The transmitter transmits each bit for a duration of 16 SCI baud clock periods. During the first clock period for a bit, the transmitter shifts the value of that bit onto the SCITX pin. The transmitter then holds the current bit value on SCITX for 16 SCI baud clock periods.



**Figure 15-3. Asynchronous Communication Bit Timing**

### 15.2.2.2 Isosynchronous Timing Mode

In isosynchronous timing mode, each bit in a frame has a duration of exactly 1 baud clock period and therefore consists of a single sample. With this timing configuration, the transmitter and receiver are required to make use of the SCICLK pin to synchronize communication with other SCI. **This mode is not fully supported on this device because SCICLK pin is not available.**



### 15.2.3 SCI Baud Rate

The SCI has an internally generated serial clock determined by the peripheral VCLK and the prescalers BAUD. The SCI uses the 24-bit integer prescaler BAUD value of the BRS register to select the required baud rates.

In asynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$\text{Asynchronous baud value} = \frac{\text{VCLK Frequency}}{16 * (\text{BAUD} + 1)}$$

For BAUD = 0,

$$\text{Asynchronous baud value} = \frac{\text{VCLK Frequency}}{32} \quad (3)$$

In isosynchronous timing mode, the SCI generates a baud clock according to the following formula:

$$\text{Isosynchronous baud value} = \frac{\text{VCLK Frequency}}{\text{BAUD} + 1}$$

For BAUD = 0,

$$\text{Isosynchronous baud value} = \frac{\text{VCLK Frequency}}{32} \quad (4)$$

### 15.2.4 SCI Multiprocessor Communication Modes

In some applications, the SCI may be connected to more than one serial communication device. In such a multiprocessor configuration, several frames of data may be sent to all connected devices or to an individual device. In the case of data sent to an individual device, the receiving devices must determine when they are being addressed. When a message is not intended for them, the devices can ignore the following data. When only two devices make up the SCI network, addressing is not needed, so multiprocessor communication schemes are not required.

SCI supports two multiprocessor Communication Modes which can be selected using COMM MODE bit:

- Idle-Line Mode
- Address Bit Mode

When the SCI is not used in a multiprocessor environment, software can consider all frames as data frames. In this case, the only distinction between the idle-line and address-bit modes is the presence of an extra bit (the address bit) in each frame sent with the address-bit protocol.

The SCI allows full-duplex communication where data can be sent and received via the transmit and receive pins simultaneously. However, the protocol used by the SCI assumes that only one device transmits data on the same bus line at any one time. No arbitration is done by the SCI.

---

#### Note

#### Avoid Transmitting Simultaneously on the Same Serial Bus

The system designer must ensure that devices connected to the same serial bus line do not attempt to transmit simultaneously. If two devices are transmitting different data, the resulting bus conflict could damage the device..

---



#### 15.2.4.2 Address-Bit Multiprocessor Mode

In the address-bit protocol, each frame has an extra bit immediately following the data field called an address bit. A frame with the address bit set to 1 is an address frame; a frame with the address bit set to 0 is a data frame. The idle period timing is irrelevant in this mode. Figure 15-5 illustrates the format of several blocks and frames with the address-bit mode.

When address-bit mode is used, the value of the TXWAKE bit is the value sent as the address bit. To send an address frame, software must set the TXWAKE bit. This bit is cleared as the contents of the SCITD are shifted from the TXWAKE register so that all frames sent are data except when the TXWAKE bit is written as a 1.

No dummy write to SCITD is required before an address frame is sent in address-bit mode. The first byte written to SCITD after the TXWAKE bit is written to 1 is transmitted with the address bit set when address-bit mode is used.

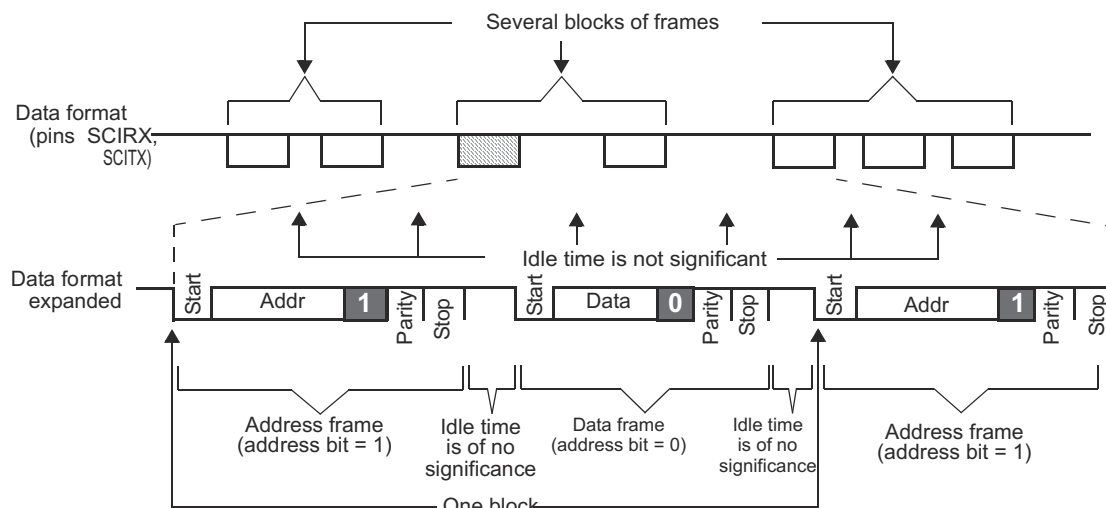


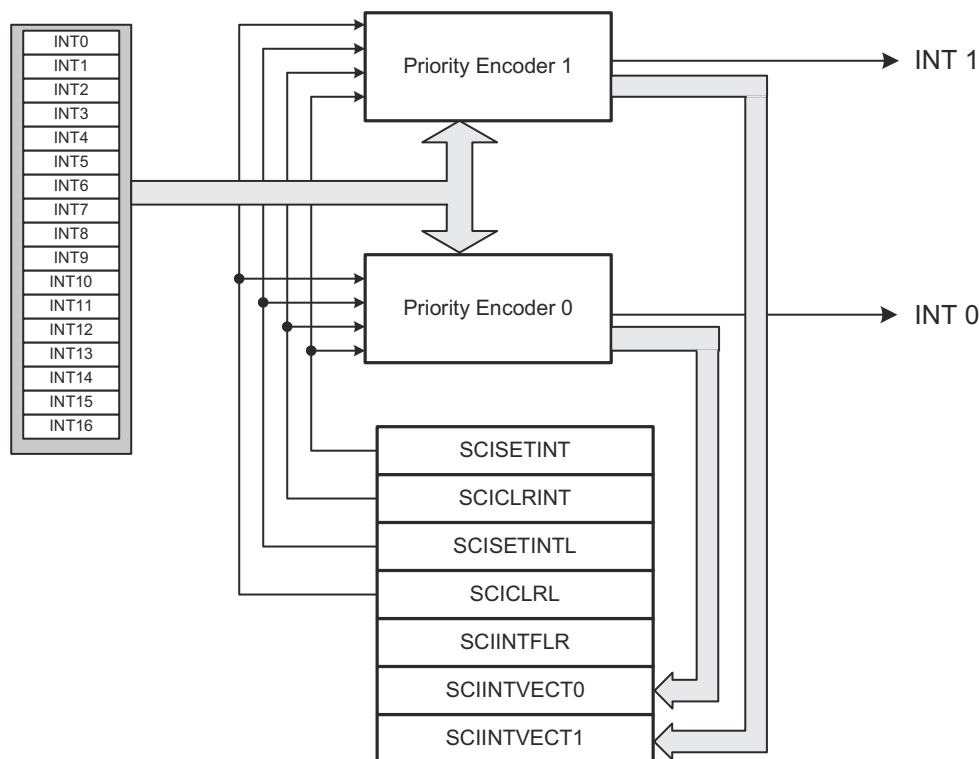
Figure 15-5. Address-Bit Multiprocessor Communication Format

## 15.3 SCI Interrupts

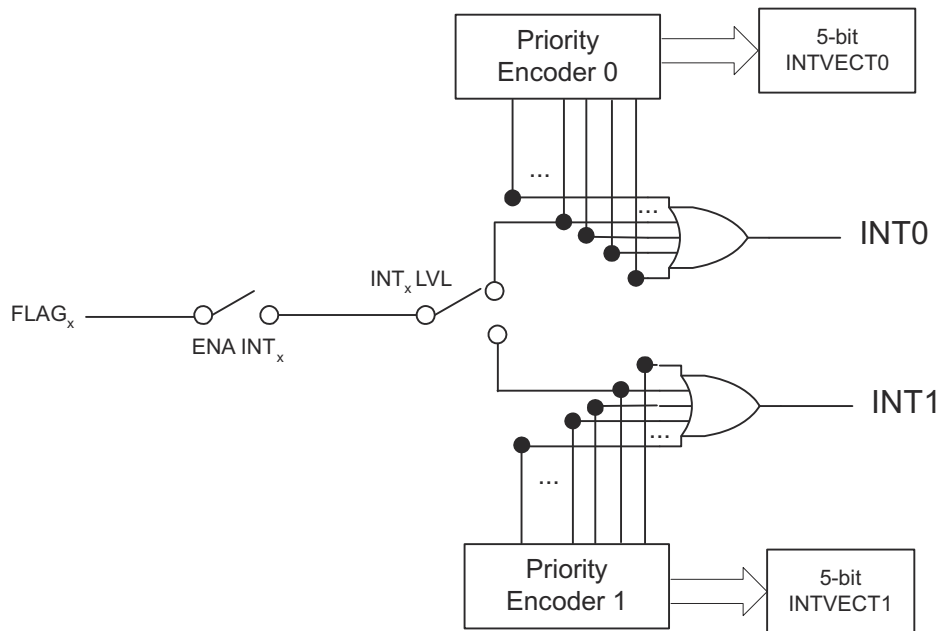
The SCI module has two interrupt lines, level 0 and level 1, to the interrupt manager (NVIC/IM) module (see [Figure 15-6](#)). Two offset registers SCIINTVECT0 and SCIINTVECT1 determine which flag triggered the interrupt according to the respective priority encoders. Each interrupt condition has a bit to enable and disable the interrupt in the SCISSETINT and SCICLRINT registers, respectively.

Each interrupt also has a bit that can be set as interrupt level 0 (INT0) or as interrupt level 1 (INT1). By default, interrupts are in interrupt level 0. SCISSETINTLVL sets a given interrupt to level1. SCICLEARINTLVL resets a given interrupt level to the default level 0.

The interrupt vector registers SCIINTVECT0 and SCIINTVECT1 return the vector of the pending interrupt line INT0 or INT1. If more than one interrupt is pending, the interrupt vector register holds the highest priority interrupt.



**Figure 15-6. General Interrupt Scheme**



**Figure 15-7. Interrupt Generation for Given Flags**

### 15.3.1 Transmit Interrupt

To use transmit interrupt functionality, SET TX INT bit must be enabled and SET TX DMA bit must be cleared. The transmit ready (TXRDY) flag is set when the SCI transfers the contents of SCITD to the shift register, SCITXSHF. The TXRDY flag indicates that SCITD is ready to be loaded with more data. In addition, the SCI sets the TX EMPTY bit if both the SCITD and SCITXSHF registers are empty. If the SET TX INT bit is set, then a transmit interrupt is generated when the TXRDY flag goes high. Transmit Interrupt is not generated immediately after setting the SET TX INT bit unlike transmit DMA request. Transmit Interrupt is generated only after the first transfer from SCITD to SCITXSHF, that is first data has to be written to SCITD by the User before any interrupt gets generated. To transmit further data the user can write data to SCITD in the transmit Interrupt service routine.

Writing data to the SCITD register clears the TXRDY bit. When this data has been moved to the SCITXSHF register, the TXRDY bit is set again. The interrupt request can be suspended by setting the CLR TX INT bit; however, when the SET TX INT bit is again set to 1, the TXRDY interrupt is asserted again. The transmit interrupt request can be eliminated until the next series of values is written to SCITD, by disabling the transmitter via the TXENA bit, by a software reset SWnRST, or by a device hardware reset.

### 15.3.2 Receive Interrupt

The receive ready (RXRDY) flag is set when the SCI transfers newly received data from SCIRXSHF to SCIRD. The RXRDY flag therefore indicates that the SCI has new data to be read. Receive interrupts are enabled by the SET RX INT bit. If the SET RX INT is set when the SCI sets the RXRDY flag, then a receive interrupt is generated. The received data can be read in the Interrupt Service routine.

On a device with both SCI and a DMA controller, the bits SET RX DMA ALL and SET RX DMA must be cleared to select interrupt functionality.

### 15.3.3 WakeUp Interrupt

SCI sets the WAKEUP flag if bus activity on the RX line either prevents power-down mode from being entered, or RX line activity causes an exit from power-down mode. If enabled (SET WAKEUP INT), wakeup interrupt is triggered once WAKEUP flag is set.

### 15.3.4 Error Interrupts

The following error detection features are supported with Interrupt by the SCI module:

- Parity errors (PE)
- Frame errors (FE)
- Break Detect errors (BRKDT)
- Overrun errors (OE)

If any of these errors (PE, FE, BRKDT, OE) is flagged, an interrupt for the flagged errors will be generated if enabled. A message is valid for both the transmitter and the receiver if there is no error detected until the end of the frame. Each of these flags is located in the receiver status (SCIFLR) register. Further details on these flags are explained in SCIFLR register description.

The SCI module supports the following 7 interrupts as listed in [Table 15-1](#).

**Table 15-1. SCI Interrupts**

Offset <sup>(1)</sup>	Interrupt
0	Reserved
1	Wakeup
2	Reserved
3	Parity error
4	Reserved
5	Reserved
6	Frame error
7	Break detect error
8	Reserved
9	Overrun error
10	Reserved
11	Receive
12	Transmit
13-15	Reserved

(1) Offset 1 is the highest priority. Offset 16 is the lowest priority.

## 15.4 SCI DMA Interface

DMA requests for receive (RXDMA request) and transmit (TXDMA request) are available for the SCI module. Refer to the DMA module chapter for DMA module configurations.

### 15.4.1 Receive DMA Requests

This DMA functionality is enabled/disabled by the CPU using the SET RX DMA/CLR RX DMA bits, respectively.

The receiver DMA request is set when a frame is received successfully and DMA functionality has been previously enabled. The RXRDY flag is set when the SCI transfers newly received data from the SCIRXSHF register to the SCIRD buffer. The RXRDY flag therefore indicates that the SCI has new data to be read. Receive DMA requests are enabled by the SET RX INT bit.

Parity, overrun, break detect, wakeup, and framing errors generate an error interrupt request immediately upon detection, if enabled, even if the device is in the process of a DMA data transfer. The DMA transfer is postponed until the error interrupt is served. The error interrupt can delete this particular DMA request by reading the receive buffer.

In multiprocessor mode, the SCI can generate receiver interrupts for address frames and DMA requests for data frames. This is controlled by an extra select bit SET RX DMA ALL.

If the SET RX DMA ALL bit is set and the SET RX DMA bit is set when the SCI sets the RXRDY flag, then a receive DMA request is generated for address and data frames.

If the SET RX DMA ALL bit is cleared and the SET RX DMA bit is set when the SCI sets the RXRDY flag upon receipt of a data frame, then a receive DMA request is generated. Receive interrupt requests are generated for address frames.

In multiprocessor mode with the SLEEP bit set, no DMA is generated for received data frames. The software must clear the SLEEP bit before data frames can be received. [Table 15-2](#) specifies the bit values for DMA requests in multiprocessor modes.

**Table 15-2. DMA and Interrupt Requests in Multiprocessor Modes**

SET RX INT	SET RX DMA	SET RX DMA ALL	ADDR FRAME INT	ADDR FRAME DMA	DATA FRAME INT	DATA FRAME DMA
0	0	x	N	N	N	N
0	1	0	Y	N	N	Y
0	1	1	N	Y	N	Y
1	0	x	Y	N	Y	N
1	1	0	Y	N	Y	Y
1	1	1	Y	Y	Y	Y

In multiprocessor mode, the SCI can generate receiver interrupts for address frames and DMA requests for data frames or DMA requests for both. This is controlled by the SET RX DMA ALL bit.

In multiprocessor mode with the SLEEP bit set, no DMA is generated for received data frames. The software must clear the SLEEP bit before data frames can be received.

### 15.4.2 Transmit DMA Requests

DMA functionality is enabled and disabled by the CPU with the SET TX DMA and CLR TX DMA bits, respectively.

The TXRDY flag is set when the SCI transfers the contents of SCITD to SCITXSHF. The TXRDY flag indicates that SCITD is ready to be loaded with more data. In addition, the SCI sets the TX EMPTY bit if both the SCITD and SCITXSHF registers are empty.

Transmit DMA requests are enabled by the setting SET TX DMA and SET TX INT bits. If the SET TX DMA bit is set, then a TX DMA request is sent to the DMA when data is written to SCITD and TXRDY is set. The DMA will write the first byte to the transmit buffer.



## 15.5 SCI Configurations

Before the SCI sends or receives data, its registers should be properly configured. Upon power-up or a system-level reset, each bit in the SCI registers is set to a default state. The registers are writable only after the RESET bit in the SCIGCR0 register is set to 1. Of particular importance is the SWnRST bit in the SCIGCR1 register. The SWnRST is an active-low bit initialized to 0 and keeps the SCI in a reset state until it is programmed to 1. Therefore, all SCI configuration should be completed before a 1 is written to the SWnRST bit.

The following list details the configuration steps that software should perform prior to the transmission or reception of data. As long as the SWnRST bit is cleared to 0 the entire time that the SCI is being configured, the order in which the registers are programmed is not important.

- Enable SCI by setting the RESET bit to 1.
- Clear the SWnRST bit to 0 before SCI is configured.
- Select the desired frame format by programming the SCIGCR1 register.
- Set both the RX FUNC and TX FUNC bits in SCIPIO0 to 1 to configure the SCIRX and SCITX pins for SCI functionality.
- Select the baud rate to be used for communication by programming the BRS register.
- Set the CLOCK bit in SCIGCR1 to 1 to select the internal clock.
- Set the CONT bit in SCIGCR1 to 1 to make SCI not halt for an emulation breakpoint until its current reception or transmission is complete (this bit is used only in an emulation environment).
- Set LOOP BACK bit in SCIGCR1 to 1 to connect the transmitter to the receiver internally (this feature is used to perform a self-test).
- Set the RXENA bit in SCIGCR1 to 1, if data is to be received.
- Set the TXENA bit in SCIGCR1 to 1, if data is to be transmitted.
- Set the SWnRST bit to 1 after SCI is configured.
- Perform receiving or transmitting data (see [Section 15.5.1](#) and [Section 15.5.2](#)).

### 15.5.1 Receiving Data

The SCI receiver is enabled to receive messages if both the RX FUNC bit and the RXENA bit are set to 1. If the RX FUNC bit is not set, the SCIRX pin functions as a general-purpose I/O pin rather than as an SCI function pin. After a valid idle period is detected, data is automatically received as it arrives on the SCIRX pin.

SCI sets the RXRDY bit when it transfers newly received data from SCIRXSHF to SCIRD. The SCI clears the RXRDY bit after the new data in SCIRD has been read. Also, as data is transferred from SCIRXSHF to SCIRD, the SCI sets the FE, OE, or PE flags if any of these error conditions were detected in the received data. These error conditions are supported with configurable interrupt capability. The wakeup and break-detect status bits are also set if one of these errors occurs, but they do not necessarily occur at the same time that new data is being loaded into SCIRD.

You can receive data by:

1. Polling Receive Ready Flag
2. Receive Interrupt
3. DMA

In polling method, software can poll for the RXRDY bit and read the data from SCIRD register once RXRDY is set high. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use either the interrupt or DMA method. To use the interrupt method, the SET RX INT bit is set. To use the DMA method, the SET RX DMA bit is set. Either an interrupt or a DMA request is generated the moment the RXRDY bit is set.

### 15.5.2 Transmitting Data

The SCI transmitter is enabled if both the TX FUNC bit and the TXENA bit are set to 1. If the TX FUNC bit is not set, the SCITX pin functions as a general-purpose I/O pin rather than as an SCI function pin. Any value written to the SCITD before TXENA is set to 1 is not transmitted. Both of these control bits allow for the SCI transmitter to be held inactive independently of the receiver.

SCI waits for data to be written to SCITD, transfers it to SCITXSHF, and transmits the data. The TXRDY and TX EMPTY bits indicate the status of the transmit buffers. That is, when the transmitter is ready for data to be written to SCITD, the TXRDY bit is set. Additionally, if both SCITD and SCITXSHF are empty, then the TX EMPTY bit is also set.

You can transmit data by:

1. Polling Transmit Ready Flag
2. Transmit Interrupt
3. DMA

In polling method, software can poll for the TXRDY bit to go high before writing the data to the SCITD register. The CPU is unnecessarily overloaded by selecting the polling method. To avoid this, you can use either the interrupt or DMA method. To use the interrupt method, the SET TX INT bit is set. To use the DMA method, the SET TX DMA bit is set. Either an interrupt or a DMA request is generated the moment the TXRDY bit is set. When the SCI has completed transmission of all pending frames, the SCITXSHF register and SCITD are empty, the TXRDY bit is set, and an interrupt/DMA request is generated, if enabled. Because all data has been transmitted, the interrupt/DMA request should be halted. This can either be done by disabling the transmit interrupt (CLR TX INT) / DMA request (CLR TX DMA bit) or by disabling the transmitter (clear TXENA bit).

---

#### Note

The TXRDY flag cannot be cleared by reading the corresponding interrupt offset in the SCIINTVECT0 or SCIINTVECT1 register.

---

## 15.6 MSS\_SCI Registers

Table 15-3 lists the memory-mapped registers for the MSS\_SCI registers. All register offset addresses not listed in Table 15-3 should be considered as reserved locations and the register contents should not be modified.

**Table 15-3. MSS\_SCI Registers**

Offset	Acronym	Register Name	Section
0h	SCIGCR0	The SCIGCR0 register defines the module reset	<a href="#">Go</a>
4h	SCIGCR1	The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI	<a href="#">Go</a>
8h	RESERVED1	Reserved	<a href="#">Go</a>
Ch	SCISETINT	SCI Set Interrupt Register	<a href="#">Go</a>
10h	SCICLEARINT	SCI Clear Interrupt Register	<a href="#">Go</a>
14h	SCISETINTLVL	SCI Set Interrupt Level Register	<a href="#">Go</a>
18h	SCICLEARINTLVL	SCI Clear Interrupt Level Register	<a href="#">Go</a>
1Ch	SCIFLR	SCI Flags Register	<a href="#">Go</a>
20h	SCIINTVECT0	SCI Interrupt Offset Vector 0 Register	<a href="#">Go</a>
24h	SCIINTVECT1	SCI Interrupt Offset Vector 1 Register	<a href="#">Go</a>
28h	SCICHR	SCI Character Control Register	<a href="#">Go</a>
2Ch	SCIBAUD	SCI Baud Rate Selection Register	<a href="#">Go</a>
30h	SCIED	Receiver Emulation Data Buffer	<a href="#">Go</a>
34h	SCIRD	Receiver Data Buffer	<a href="#">Go</a>
38h	SCITD	Transmit Data Buffer Register	<a href="#">Go</a>
3Ch	SCIPIO0	SCI Pin I/O Control Register 0	<a href="#">Go</a>
40h	SCIPIO1	SCI Pin I/O Control Register 1	<a href="#">Go</a>
44h	SCIPIO2	SCI Pin I/O Control Register 2	<a href="#">Go</a>
48h	SCIPIO3	SCI Pin I/O Control Register 3	<a href="#">Go</a>
4Ch	SCIPIO4	SCI Pin I/O Control Register 4	<a href="#">Go</a>
50h	SCIPIO5	SCI Pin I/O Control Register 5	<a href="#">Go</a>
54h	SCIPIO6	SCI Pin I/O Control Register 6	<a href="#">Go</a>
58h	SCIPIO7	SCI Pin I/O Control Register 7	<a href="#">Go</a>
5Ch	SCIPIO8	SCI Pin I/O Control Register 8	<a href="#">Go</a>
60h	RESERVED2	Reserved	<a href="#">Go</a>
64h	RESERVED3	Reserved	<a href="#">Go</a>
68h	RESERVED4	Reserved	<a href="#">Go</a>
6Ch	RESERVED5	Reserved	<a href="#">Go</a>
70h	RESERVED6	Reserved	<a href="#">Go</a>
74h	RESERVED7	Reserved	<a href="#">Go</a>
78h	RESERVED8	Reserved	<a href="#">Go</a>
7Ch	RESERVED9	Reserved	<a href="#">Go</a>
80h	SCIPIO9	SCI Pin I/O Control Register 9	<a href="#">Go</a>
90h	SCIIOCTRL	SCI IO DFT Control	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. Table 15-4 shows the codes that are used for access types in this section.

**Table 15-4. MSS\_SCI Access Type Codes**

Access Type	Code	Description
Read Type		

**Table 15-4. MSS\_SCI Access Type Codes  
(continued)**

Access Type	Code	Description
R	R	Read
<b>Write Type</b>		
W	W	Write
<b>Reset or Default Value</b>		
-n		Value after reset or the default value

### 15.6.1 SCIGCR0 Register (Offset = 0h) [Reset = 00000000h]

SCIGCR0 is shown in [Table 15-5](#).

Return to the [Summary Table](#).

The SCIGCR0 register defines the module reset

**Table 15-5. SCIGCR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	Reserved
0	RESET	R/W	0h	GIO reset

### 15.6.2 SCIGCR1 Register (Offset = 4h) [Reset = 00000000h]

SCIGCR1 is shown in [Table 15-6](#).

Return to the [Summary Table](#).

The SCIGCR1 register defines the frame format, protocol, and communication mode used by the SCI

**Table 15-6. SCIGCR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	Reserved
25	TXENA	R/W	0h	Data is transferred from SCITD to SCITXSHF only when the TXENA bit is set
24	RXENA	R/W	0h	Allows the receiver to transfer data from the shift buffer to the receive buffer
23-18	RESERVED	R	0h	Reserved
17	CONT	R/W	0h	This bit has an effect only when a program is being debugged with an emulator, and it determines how the SCI operates when the program is suspended
16	LOOP_BACK	R/W	0h	Enable bit for loopback mode
15-10	RESERVED	R	0h	Reserved
9	POWERDOWN	R/W	0h	When the POWERDOWN bit is set, the SCI attempts to enter local low-power mode
8	SLEEP	R/W	0h	In a multiprocessor configuration, this bit controls the receive sleep function. Clearing this bit brings the SCI out of sleep mode
7	SW_nRESET	R/W	0h	Software reset (active low)
6	RESERVED	R	0h	Reserved
5	CLOCK	R/W	0h	SCI internal clock enable
4	STOP	R/W	0h	SCI number of stop bits
3	PARITY	R/W	0h	SCI parity odd/even selection
2	PARITY_ENA	R/W	0h	SCI parity enable
1	TIMING_MODE	R/W	0h	SCI timing mode bit (0=Isosynchronous timing, 1=Asynchronous timing)
0	COMM_MODE	R/W	0h	SCI communication mode bit (0=Idle-line mode, 1=Address-bit mode)

### 15.6.3 RESERVED1 Register (Offset = 8h) [Reset = 00000000h]

RESERVED1 is shown in [Table 15-7](#).

Return to the [Summary Table](#).

Reserved

**Table 15-7. RESERVED1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.4 SCISSETINT Register (Offset = Ch) [Reset = 00000000h]

SCISSETINT is shown in [Table 15-8](#).

Return to the [Summary Table](#).

SCI Set Interrupt Register

**Table 15-8. SCISSETINT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	SET_FE_INT	R/W	0h	Set Framing-Error Interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
25	SET_OE_INT	R/W	0h	Set Overrun-Error Interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
24	SET_PE_INT	R/W	0h	Set Parity Interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
23-19	RESERVED	R	0h	Reserved
18	SET_RX_DMA_ALL	R/W	0h	Determines if a separate interrupt is generated for the address frames sent in multiprocessor communications User and privilege mode (read): 0 = DMA request is disabled for address frames (RX interrupt request is enabled for address frames) 1 = DMA request is enabled for address and data frames User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request for address and data frames
17	SET_RX_DMA	R/W	0h	To select receiver DMA requests, this bit must be set. If it is cleared, interrupt requests are generated depending on bit SCISSETINT.9 User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request
16	SET_TX_DMA	R/W	0h	To select DMA requests for the transmitter, this bit must be set. If it is cleared, interrupt requests are generated depending on SET TX INT bit (SCISSETINT.8) User and privilege mode (read): 0 = TX interrupt request selected 1 = TX DMA request selected User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
15-10	RESERVED	R	0h	Reserved
9	SET_RX_INT	R/W	0h	Receiver interrupt enable: Setting this bit enables the SCI to generate a receive interrupt after a frame has been completely received and the data is being transferred from SCIRXSHF to SCIRD. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt



**Table 15-8. SCISSETINT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	SET_TX_INT	R/W	0h	Set Transmitter interrupt. Setting this bit enables the SCI to generate a transmit interrupt as data is being transferred from SCITD to SCITXSHF and the TXRDY bit is being set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
7-2	RESERVED	R	0h	Reserved
1	SET_WAKEUP_INT	R/W	0h	Set Wake-up interrupt User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
0	SET_BRKDT_INT	R/W	0h	Set Break-detect interrupt. Setting this bit enables the SCI to generate an error interrupt if a break condition is detected on the SCIRX pin. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt

### 15.6.5 SCICLEARINT Register (Offset = 10h) [Reset = 00000000h]

SCICLEARINT is shown in [Table 15-9](#).

Return to the [Summary Table](#).

SCI Clear Interrupt Register

**Table 15-9. SCICLEARINT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	CLR_FE_INT	R/W	0h	Clear Framing-Error Interrupt: Setting this bit disables the SCI module to generate an interrupt when there is a Framing error. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
25	CLR_OE_INT	R/W	0h	Clear Overrun-Error Interrupt. This bit disables the SCI overrun interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
24	CLR_PE_INT	R/W	0h	Clear Parity Interrupt. Setting this bit disables the SCI Parity error interrupt. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt
23-19	RESERVED	R	0h	Reserved
18	CLR_RX_DMA_ALL	R/W	0h	User and privilege mode (read): 0 = DMA request is disabled for address frames (RX interrupt request is enabled for address frames). DMA request is enabled for data frames. 1 = DMA request is enabled for address and data frames User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request for address frames
17	CLR_RX_DMA	R/W	0h	Clear RX DMA request. This bit disables the receive DMA request when set. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
16	CLR_TX_DMA	R/W	0h	Clear TX DMA request. This bit disables the transmit DMA request when set. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request
15-10	RESERVED	R	0h	Reserved
9	CLR_RX_INT	R/W	0h	Clear Receiver interrupt. This bit disables the receiver interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

**Table 15-9. SCICLEARINT Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	CLR_TX_INT	R/W	0h	Clear Transmitter interrupt. This bit disables the transmitter interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
7-2	RESERVED	R	0h	Reserved
1	CLR_WAKEUP_INT	R/W	0h	Clear Wake-up interrupt. This bit disables the wakeup interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt
0	CLR_BRKDT_INT	R/W	0h	Clear Break-detect interrupt. This bit disables the Break-detect interrupt when set. User and privilege mode (read): 0 = Interrupt is disabled 1 = Interrupt is enabled User and privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt

### 15.6.6 SCISSETINTLVL Register (Offset = 14h) [Reset = 00000000h]

SCISSETINTLVL is shown in [Table 15-10](#).

Return to the [Summary Table](#).

SCI Set Interrupt Level Register

**Table 15-10. SCISSETINTLVL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	SET_FE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
25	SET_OE_INT_LVL	R/W	0h	Clear Overrun-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
24	SET_PE_INT_LVL	R/W	0h	Clear Parity Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
23-19	RESERVED	R	0h	Reserved
18	SET_RX_DMA_ALL_INT_LVL	R/W	0h	User and privilege mode (read): 0 = RX interrupt request for address frames mapped to INT0 line. 1 = RX interrupt request for address frames mapped to INT1 line. User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
17-16	RESERVED	R	0h	Reserved
15	SET_INC_BR_INT_LVL	R/W	0h	
14-10	RESERVED	R	0h	Reserved
9	SET_RX_INT_LVL	R/W	0h	Clear Receiver interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
8	SET_TX_INT_LVL	R/W	0h	Clear Transmitter interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
7-2	RESERVED	R	0h	Reserved

**Table 15-10. SCISSETINTLVL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	SET_WAKEUP_INT_LVL	R/W	0h	Clear Wake-up interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1
0	SET_BRKDT_INT_LVL	R/W	0h	Clear Break-detect interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Clear interrupt level to line INT1

### 15.6.7 SCICLEARINTLVL Register (Offset = 18h) [Reset = 00000000h]

SCICLEARINTLVL is shown in [Table 15-11](#).

Return to the [Summary Table](#).

SCI Clear Interrupt Level Register

**Table 15-11. SCICLEARINTLVL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	CLR_FE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
25	CLR_OE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
24	CLR_PE_INT_LVL	R/W	0h	Clear Framing-Error Interrupt Level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
23-19	RESERVED	R	0h	Reserved
18	CLR_RX_DMA_ALL_INT_LVL	R/W	0h	Clear receive DMA ALL interrupt level. User and privilege mode (read): 0 = RX interrupt request for address frames is mapped to INT0 line. 1 = RX interrupt request for address frames is mapped to INT1 line. User and privilege mode (write): 0 = Leaves the corresponding bit unchanged. 1 = Reset interrupt level to line INT0.
17-16	RESERVED	R	0h	Reserved
15	CLR_INC_BR_INT_LVL	R/W	0h	
14-10	RESERVED	R	0h	Reserved
9	CLR_RX_INT_LVL	R/W	0h	Clear Receiver interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
8	CLR_TX_INT_LVL	R/W	0h	Clear Transmitter interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
7-2	RESERVED	R	0h	Reserved

**Table 15-11. SCICLEARINTLVL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
1	CLR_WAKEUP_INT_LVL	R/W	0h	Clear Wake-up interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0
0	CLR_BRKDT_INT_LVL	R/W	0h	Clear Break-detect interrupt level. User and privilege mode (read): 0 = Interrupt level mapped to INT0 line 1 = Interrupt level mapped to INT1 line User and privilege mode (write): 0 = Leaves the corresponding bit unchanged 1 = Reset interrupt level to line INT0

### 15.6.8 SCIFLR Register (Offset = 1Ch) [Reset = 00000904h]

SCIFLR is shown in [Table 15-12](#).

Return to the [Summary Table](#).

SCI Flags Register

**Table 15-12. SCIFLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	FE	R/W	0h	SCI framing error flag Read: 0=No framing error detected 1=Framing error detected Write: 0=No effect 1=Clears this bit to 0
25	OE	R	0h	SCI overrun error flag This bit is set when the transfer of data from SCIRXSHF to SCIRD overwrites unread data already in SCIRD
24	PE	R	0h	SCI parity error flag. This bit is set when a parity error is detected in the received data
23-13	RESERVED	R	0h	Reserved
12	RXWAKE	R	0h	Receiver wake-up detect flag. The SCI sets this bit to indicate that the data currently in SCIRD is an address
11	TX_EMPTY	R	1h	Transmitter empty flag. The value of this flag indicates the contents of the transmitter's buffer register (SCITD) and shift register (SCITXSHF)
10	TXWAKE	R/W	0h	SCI transmitter wake-up method select. The TXWAKE bit controls whether the data in SCITD should be sent as an address or data frame using multiprocessor communication format
9	RXRDY	R	0h	SCI receiver ready flag. The receiver sets this bit to indicate that the SCIRD contains new data and is ready to be read by the CPU or DMA.
8	TXRDY	R	1h	Transmitter buffer register ready flag. When set, this bit indicates that the transmit buffer register (SCITD) is ready to receive another character.
7-4	RESERVED	R	0h	Reserved
3	Bus_busy_flag	R	0h	This bit indicates whether the receiver is in the process of receiving a frame.
2	IDLE	R	1h	SCI receiver in idle state. While this bit is set, the SCI looks for an idle period to resynchronize itself with the bit stream.
1	WAKEUP	R	0h	Wake-up flag. This bit is set by the SCI when receiver or transmitter activity has taken the module out of power-down mode.
0	BRKDT	R	0h	SCI break-detect flag. This bit is set when the SCI detects a break condition on the SCIRX pin.



### 15.6.9 SCIINTVECT0 Register (Offset = 20h) [Reset = 00000000h]

SCIINTVECT0 is shown in [Table 15-13](#).

Return to the [Summary Table](#).

SCI Interrupt Offset Vector 0 Register

**Table 15-13. SCIINTVECT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	INTVECT0	R	0h	Interrupt vector offset for INT0

### 15.6.10 SCIINTVECT1 Register (Offset = 24h) [Reset = 00000000h]

SCIINTVECT1 is shown in [Table 15-14](#).

Return to the [Summary Table](#).

SCI Interrupt Offset Vector 1 Register

**Table 15-14. SCIINTVECT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R	0h	Reserved
3-0	INTVECT1	R	0h	Interrupt vector offset for INT1

### 15.6.11 SCICCHAR Register (Offset = 28h) [Reset = 00000000h]

SCICCHAR is shown in [Table 15-15](#).

Return to the [Summary Table](#).

SCI Character Control Register

**Table 15-15. SCICCHAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2-0	CHAR	R/W	0h	Sets the SCI data length from 1 to 8 bits

### 15.6.12 SCIBAUD Register (Offset = 2Ch) [Reset = 00000000h]

SCIBAUD is shown in [Table 15-16](#).

Return to the [Summary Table](#).

SCI Baud Rate Selection Register

**Table 15-16. SCIBAUD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	Reserved
23-0	BAUD	R/W	0h	SCI 24-bit baud selection

### 15.6.13 SCIED Register (Offset = 30h) [Reset = 00000000h]

SCIED is shown in [Table 15-17](#).

Return to the [Summary Table](#).

Receiver Emulation Data Buffer

**Table 15-17. SCIED Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	ED	R	0h	Receiver Emulation Data Buffer

### 15.6.14 SCIRD Register (Offset = 34h) [Reset = 00000000h]

SCIRD is shown in [Table 15-18](#).

Return to the [Summary Table](#).

Receiver Data Buffer

**Table 15-18. SCIRD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	RD	R	0h	Contains received data.

### 15.6.15 SCITD Register (Offset = 38h) [Reset = 00000000h]

SCITD is shown in [Table 15-19](#).

Return to the [Summary Table](#).

Transmit Data Buffer Register

**Table 15-19. SCITD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	Reserved
7-0	TD	R/W	0h	Contains Data to be transmitted. This is pushed to SCITXSHF(shift register) when TXENA bit is set in SCRGCR1 register.

### 15.6.16 SCIPIO0 Register (Offset = 3Ch) [Reset = 00000000h]

SCIPIO0 is shown in [Table 15-20](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 0

**Table 15-20. SCIPIO0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_FUNC	R/W	0h	Defines the function of pin SCITX. 0=SCITX is a general-purpose digital I/O pin. 1=SCITX is the SCI transmit pin. 0=SCIRX is a general-purpose digital I/O pin. 1=SCIRX is the SCI receive pin.
1	RX_FUNC	R/W	0h	Determines the data direction on the SCIRX pin if it is configured with general-purpose I/O functionality (RX_FUNC = 0). See Table 12 for bit values. 0=SCIRX is a general-purpose input pin. 1=SCIRX is a general-purpose output pin
0	CLK_FUNC	R/W	0h	Clock function. Defines the function of pin SCICLK. 0=SCICLK is a general-purpose digital I/O pin. 1=SCICLK is the SCI serial clock pin. Determines the data direction on the SCICLK pin. The direction is defined differently depending upon the value of the CLK_FUNC bit



### 15.6.17 SCIPIO1 Register (Offset = 40h) [Reset = 00000000h]

SCIPIO1 is shown in [Table 15-21](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 1

**Table 15-21. SCIPIO1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DIR	R/W	0h	Determines the data direction on the SCITX pin if it is configured with general-purpose I/O functionality (TX FUNC = 0). See Table 11 for bit values. 0=SCITX is a general-purpose input pin. 1=SCITX is a general-purpose output pin
1	RX_DIR	R/W	0h	Determines the data direction on the SCIRX pin if it is configured with general-purpose I/O functionality (RX FUNC = 0). See Table 12 for bit values. 0=SCIRX is a general-purpose input pin. 1=SCIRX is a general-purpose output pin
0	CLK_DIR	R/W	0h	Clock data direction. Determines the data direction on the SCICLK pin. The direction is defined differently depending upon the value of the CLK FUNC bit

### 15.6.18 SCIPIO2 Register (Offset = 44h) [Reset = 00000000h]

SCIPIO2 is shown in [Table 15-22](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 2

**Table 15-22. SCIPIO2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_IN	R/W	0h	Contains current value on the SCITX pin. 0=SCITX value is logic low. 1=SCITX value is logic high.
1	RX_DATA_IN	R/W	0h	Contains current value on the SCIRX pin. 0=SCIRX value is logic low. 1=SCIRX value is logic high.
0	CLK_DATA_IN	R/W	0h	Contains the current value on pin SCICLK. 0=Pin SCICLK value is logic low. 1=Pin SCICLK value is logic high.

### 15.6.19 SCIPIO3 Register (Offset = 48h) [Reset = 00000000h]

SCIPIO3 is shown in [Table 15-23](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 3

**Table 15-23. SCIPIO3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_OUT	R/W	0h	Contains the data to be output on pin SCITX if the following conditions are met: TX FUNC = 0 (SCITX pin is a general-purpose I/O.) TX DATA DIR = 1 (SCITX pin is a general-purpose output.) 0=Output value on SCITX is a 0 (logic low). 1=Output value on SCITX is a 1 (logic high).
1	RX_DATA_OUT	R/W	0h	Contains the data to be output on pin SCIRX if the following conditions are met: RX FUNC = 0 (SCIRX pin is a general-purpose I/O.) RX DATA DIR = 1 (SCIRX pin is a general-purpose output.) 0=Output value on SCIRX is 0 (logic low). 1=Output value on SCIRX is 1 (logic high).
0	CLK_DATA_OUT	R/W	0h	Contains the data to be output on pin SCICLK if the following conditions are met: CLK FUNC = 0 (SCICLK pin is a general-purpose I/O.) CLK DATA DIR = 1 (SCICLK pin is a general-purpose output.) 0=Output value on SCICLK is a 0 (logic low). 1=Output value on SCICLK is a 1 (logic high).

### 15.6.20 SCIPIO4 Register (Offset = 4Ch) [Reset = 00000000h]

SCIPIO4 is shown in [Table 15-24](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 4

**Table 15-24. SCIPIO4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_SET	R/W	0h	Sets the data to be output on pin SCITX if the following conditions are met: TX FUNC = 0 (SCITX pin is a general-purpose I/O.) TX DATA DIR = 1 (SCITX pin is a general-purpose output.)
1	RX_DATA_SET	R/W	0h	Sets the data to be output on pin SCIRX if the following conditions are met: RX FUNC = 0 (SCIRX pin is a general-purpose I/O.) RX DATA DIR = 1 (SCIRX pin is a general-purpose output.)
0	CLK_DATA_SET	R/W	0h	Sets the data to be output on pin SCICLK if the following conditions are met: CLK FUNC = 0 (SCICLK pin is a general-purpose I/O.) CLK DATA DIR = 1 (SCICLK pin is a general-purpose output.)

### 15.6.21 SCIPIO5 Register (Offset = 50h) [Reset = 00000000h]

SCIPIO5 is shown in [Table 15-25](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 5

**Table 15-25. SCIPIO5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_DATA_CLR	R/W	0h	Clears the data to be output on pin SCITX if the following conditions are met: TX FUNC = 0 (SCITX pin is a general-purpose I/O.) TX DATA DIR = 1 (SCITX pin is a general-purpose output.)
1	RX_DATA_CLR	R/W	0h	Clears the data to be output on pin SCITX if the following conditions are met: TX FUNC = 0 (SCITX pin is a general-purpose I/O.) TX DATA DIR = 1 (SCITX pin is a general-purpose output.)
0	CLK_DATA_CLR	R/W	0h	Clears the data to be output on pin SCITX if the following conditions are met: TX FUNC = 0 (SCITX pin is a general-purpose I/O.) TX DATA DIR = 1 (SCITX pin is a general-purpose output.)

## 15.6.22 SCIPIO6 Register (Offset = 54h) [Reset = 00000000h]

SCIPIO6 is shown in [Table 15-26](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 6

**Table 15-26. SCIPIO6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_PDR	R/W	0h	TX Open Drain Enable Enables open-drain capability in the output pin SCITX if the following conditions are met: TX DATA DIR = 1 (SCITX pin is a general-purpose output.) TX DOUT = 1
1	RX_PDR	R/W	0h	RX Open Drain Enable Enables open-drain capability in the output pin SCIRX if the following conditions are met: RX DATA DIR = 1 (SCIRX pin is a general-purpose output.) RX DOUT = 1
0	CLK_PDR	R/W	0h	CLK Open Drain Enable Enables open-drain capability in the output pin SCICLK if the following conditions are met: CLK DATA DIR = 1 (SCICLK pin is a general-purpose output.) CLK DOUT = 1

### 15.6.23 SCIPIO7 Register (Offset = 58h) [Reset = 00000000h]

SCIPIO7 is shown in [Table 15-27](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 7

**Table 15-27. SCIPIO7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_PD	R/W	0h	TX pin Pull Control Disable Disables pull control capability in the output pin SCITX. 0=Pull Control on SCITX pin is enabled. 1=Pull Control on SCITX pin is disabled.
1	RX_PD	R/W	0h	RX pin Pull Control Disable Disables pull control capability in the output pin SCIRX. 0=Pull Control on SCIRX pin is enabled. 1=Pull Control on SCIRX pin is disabled.
0	CLK_PD	R/W	0h	CLK pin Pull Control Disable Disables pull control capability in the output pin SCICLK. 0=Pull Control on SCICLK pin is enabled. 1=Pull Control on SCICLK pin is disabled.

### 15.6.24 SCIPIO8 Register (Offset = 5Ch) [Reset = 00000000h]

SCIPIO8 is shown in [Table 15-28](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 8

**Table 15-28. SCIPIO8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_PSL	R/W	0h	TX pin Pull Select Selects pull type in the output pin SCITX. 0=Pull-Down is on SCITX pin. 1=Pull-Up is on SCITX pin.
1	RX_PSL	R/W	0h	RX pin Pull Select Selects pull type in the output pin SCIRX. 0=Pull-Down is on SCIRX pin. 1=Pull-Up is on SCIRX pin.
0	CLK_PSL	R/W	0h	CLK pin Pull Select Selects pull type in the output pin SCICLK. 0=Pull-Down is on SCICLK pin. 1=Pull-Up is on SCICLK pin.



### 15.6.25 RESERVED2 Register (Offset = 60h) [Reset = 00000000h]

RESERVED2 is shown in [Table 15-29](#).

Return to the [Summary Table](#).

Reserved

**Table 15-29. RESERVED2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.26 RESERVED3 Register (Offset = 64h) [Reset = 00000000h]

RESERVED3 is shown in [Table 15-30](#).

Return to the [Summary Table](#).

Reserved

**Table 15-30. RESERVED3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.27 RESERVED4 Register (Offset = 68h) [Reset = 00000000h]

RESERVED4 is shown in [Table 15-31](#).

Return to the [Summary Table](#).

Reserved

**Table 15-31. RESERVED4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.28 RESERVED5 Register (Offset = 6Ch) [Reset = 00000000h]

RESERVED5 is shown in [Table 15-32](#).

Return to the [Summary Table](#).

Reserved

**Table 15-32. RESERVED5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.29 RESERVED6 Register (Offset = 70h) [Reset = 00000000h]

RESERVED6 is shown in [Table 15-33](#).

Return to the [Summary Table](#).

Reserved

**Table 15-33. RESERVED6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.30 RESERVED7 Register (Offset = 74h) [Reset = 00000000h]

RESERVED7 is shown in [Table 15-34](#).

Return to the [Summary Table](#).

Reserved

**Table 15-34. RESERVED7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.31 RESERVED8 Register (Offset = 78h) [Reset = 00000000h]

RESERVED8 is shown in [Table 15-35](#).

Return to the [Summary Table](#).

Reserved

**Table 15-35. RESERVED8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved

### 15.6.32 RESERVED9 Register (Offset = 7Ch) [Reset = 00000000h]

RESERVED9 is shown in [Table 15-36](#).

Return to the [Summary Table](#).

Reserved

**Table 15-36. RESERVED9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RESERVED	R	0h	Reserved



### 15.6.33 SCIPIO9 Register (Offset = 80h) [Reset = 00000000h]

SCIPIO9 is shown in [Table 15-37](#).

Return to the [Summary Table](#).

SCI Pin I/O Control Register 9

**Table 15-37. SCIPIO9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	RESERVED	R	0h	Reserved
2	TX_SL	R/W	0h	This bit controls the slew rate for the SCITX pin. 0=The normal output buffer is used for SCITX pin 1=The output buffer with slew control is used for SCITX pin.
1	RX_SL	R/W	0h	This bit controls the slew rate for the SCIRX pin. 0=The normal output buffer is used for SCIRX pin 1=The output buffer with slew control is used for SCIRX pin
0	CLK_SL	R/W	0h	This bit controls the slew rate for the SCICLK pin. 0=The normal output buffer is used for SCICLK pin 1=The output buffer with slew control is used for SCICLK pin

### 15.6.34 SCIIODCTRL Register (Offset = 90h) [Reset = 00000000h]

SCIIODCTRL is shown in [Table 15-38](#).

Return to the [Summary Table](#).

SCI IO DFT Control

**Table 15-38. SCIIODCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	RESERVED	R	0h	Reserved
26	FEN	R/W	0h	Frame Error Enable. User and Privileged Mode Reads and Writes: 1 = This bit is used to create a Frame Error. The stop bit received is ANDed with '0' and passed to the stop bit check circuitry. 0 = No effect.
25	PEN	R/W	0h	Parity Error Enable. User and Privileged Mode Reads and Writes: 1 = This bit is used to create a Parity Error. The parity bit received is toggled so that a parity error occurs. 0 = No effect
24	BRKDT_ENA	R/W	0h	Break Detect Error Enable. User and Privileged Mode Reads and Writes: 1 = This bit is used to create BRKDT Error. The stop bit of the frame is ANDed with '0' and passed to the RSM so that a frame error occurs. Then the RX pin is forced to continuous low for 10 TBITS so that a BRKDT error occurs. 0 = No effect.
23-21	RESERVED	R	0h	Reserved
20-19	PIN_SAMPLE_MASK	R/W	0h	PIN SAMPLE MASK These bits define the sample number at which the TX Pin value that is being transmitted will be inverted to verify the receive pin samples majority detection circuitry. PIN SAMPLE MASK: 00 -- No Mask, 01 -- Invert the TX Pin value at 7th SCLK, 10 -- Invert the TX Pin value at 8th SCLK, 11 -- Invert the TX Pin value at 9th SCLK.
18-16	TX_SHIFT	R/W	0h	These bits define the delay by which the value on TX pin is delayed so that the value on RX Pin is asynchronous. (Not applicable to Start Bit) TX SHIFT: 000 -- No Delay, 001 -- Delay by 1 SCLK, 010 -- Delay by 2 SCLKs, 011 -- Delay by 3 SCLKs, 100 -- Delay by 4 SCLKs, 101 -- Delay by 5 SCLKs, 110 -- Delay by 6 SCLKs, 111 -- No Delay.
15-12	RESERVED	R	0h	Reserved
11-8	IODFTENA	R/W	0h	These bits define the delay by which the value on TX pin is delayed so that the value on RX Pin is asynchronous. (Not applicable to Start Bit) TX SHIFT: 000 -- No Delay, 001 -- Delay by 1 SCLK, 010 -- Delay by 2 SCLKs, 011 -- Delay by 3 SCLKs, 100 -- Delay by 4 SCLKs, 101 -- Delay by 5 SCLKs, 110 -- Delay by 6 SCLKs, 111 -- No Delay.
7-2	RESERVED	R	0h	Reserved

**Table 15-38. SCIIODCTRL Register Field Descriptions (continued)**

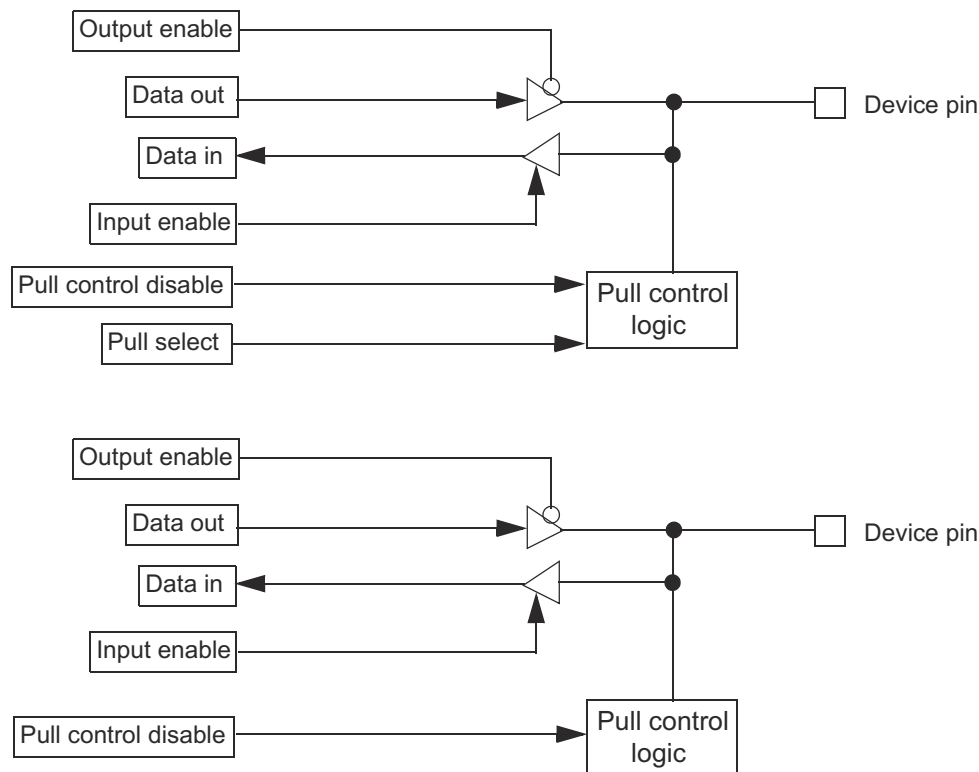
Bit	Field	Type	Reset	Description
1	LBP_ENA	R/W	0h	Module loopback enable. user and privileged mode reads: Write only in privileged mode: write/ read : 1=Analog loopback is enabled in module I/O DFT mode(when IODFTENA = 1010) 0=Digital loopback is enabled.
0	RXP_ENA	R/W	0h	Module Analog loopback through receive pin enable. user and privileged mode reads: Write only in privileged mode: write/ read : 1=Analog loopback through receive pin. 0=Analog loopback through transmit pin.

## 15.7 SCI GPIO Functionality

The following sections apply to all device pins that can be configured as functional or general-purpose I/O pins.

### 15.7.1 GPIO Functionality

Figure 15-8 illustrates the GPIO functionality.



**Figure 15-8. GPIO Functionality**

### 15.7.2 Under Reset

The following apply if a device is under reset:

- Pull control. The reset pull control on the pins is enabled.
- Input buffer. The input buffer is enabled.
- Output buffer. The output buffer is disabled.

### 15.7.3 Out of Reset

The following apply if the device is out of reset:

- Pull control. The pull control is enabled by clearing the PD (pull control disable) bit in the SCPIO7 register (SCPIO7 Register (Offset = 58h) [Reset = 00000000h] ). In this case, if the PSL (pull select) bit in the SCPIO8 register (Section 15.6.24 ) is set, the pin will have a pull-up. If the PSL bit is cleared, the pin will have a pull-down. If the PD bit is set in the control register, there is no pull-up or pull-down on the pin.
- Input buffer. The input buffer is always enabled in functional mode.

#### Note

The pull-disable logic depends on the pin direction. It is independent of whether the device is in I/O or functional mode. If the pin is configured as output or transmit, then the pulls are disabled automatically. If the pin is configured as input or receive, the pulls are enabled or disabled depending on bit PD in the pull disable register SCPIO7 (Section 15.6.23 ).

- Output buffer. A pin can be driven as an output pin if the TX DIR bit is set in the pin direction control register (SCPIO1; Section 15.6.17 ) AND the open-drain feature is not enabled in the SCPIO6 register (Section 15.6.22 ).

### 15.7.4 Open-Drain Feature Enabled on a Pin

The following apply if the open-drain feature is enabled on a pin:

- The output buffer is enabled, if a low signal is being driven on to the pin.
- The output buffer is disabled (the direction control signal DIR is internally forced low), if a high signal is being driven on to the pin.

#### Note

The open-drain feature is available only in I/O mode (SCPIO0; Section 15.6.16 ).

### 15.7.5 Summary

The behavior of the input buffer, output buffer, and the pull control is summarized in Table 15-39.

**Table 15-39. Input Buffer, Output Buffer, and Pull Control Behavior as GPIO Pins**

Device under Reset?	Pin Direction (DIR) <sup>(1) (2)</sup>	Pull Disable (PULDIS) <sup>(1) (3)</sup>	Pull Select (PULSEL) <sup>(1) (4)</sup>	Pull Control	Output Buffer	Input Buffer
Yes	X	X	X	Enabled	Disabled	Enabled
No	0	0	0	Pull down	Disabled	Enabled
No	0	0	1	Pull up	Disabled	Enabled
No	0	1	0	Disabled	Disabled	Enabled
No	0	1	1	Disabled	Disabled	Enabled
No	1	X	X	Disabled	Enabled	Enabled

(1) X = Don't care

(2) DIR = 0 for input, = 1 for output

(3) PULDIS = 0 for enabling pull control  
= 1 for disabling pull control

(4) PULSEL = 0 for pull-down functionality  
= 1 for pull-up functionality

## Chapter 16

# Inter-Integrated Circuit (I2C) Module

---



This chapter describes the inter-integrated circuit (I2C or I<sup>2</sup>C) module. The I2C is a multi-Target communication module providing an interface between the Texas Instruments (TI) microcontroller and devices compliant with Philips Semiconductor I<sup>2</sup>C-bus specification version 2.1 and connected by an I2C-bus. This module will support any Controller or Target I2C compatible device.

### 16.1 Overview

The I2C has the following features:

- Compliance to the Philips (now NXP Semiconductors) I<sup>2</sup>C bus specification, v2.1 (*The I 2 C Specification*, Philips document number 9398 393 40011)
  - Bit/Byte format transfer
  - 7-bit and 10-bit device addressing modes
  - General call
  - START byte
  - Multi-controller transmitter/target receiver mode
  - Multi-controller receiver/target transmitter mode
  - Combined controller transmit/receive and receive/transmit mode
  - Transfer rates of 10 kbps up to 400 kbps (Fast mode transfer rate)
- Free data format
- Two DMA events (transmit and receive)
- DMA event enable/disable capability
- Seven interrupts that can be used by the CPU
- Operates with VBUS frequency from 6.7 MHz up
- Operates with module frequency between 6.7 MHz and 13.3 MHz
- Module enable/disable capability
- The SDA and SCL are optionally configurable as general purpose I/O
- Slew rate control of the outputs
- Open drain control of the outputs
- Programmable pullup/pulldown capability on the inputs
- Supports Ignore NACK mode

The device has three instances of I2C called:

- MSS\_I2CA
- RCSS\_I2CA
- RCSS\_I2CB

---

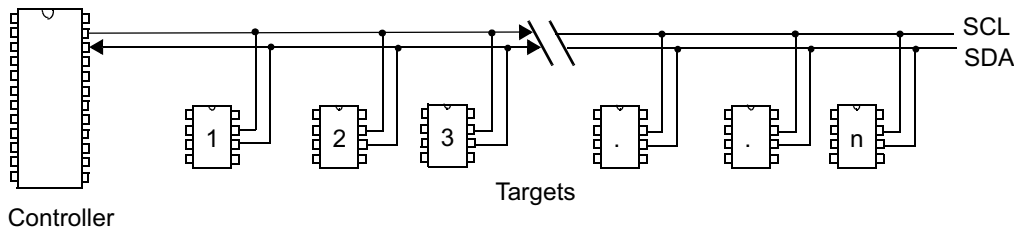
#### Note

This I2C module does **not** support:

- High-speed (HS) mode (only supports up to 400 kbps (Fast mode))
  - C-bus compatibility mode
  - The combined format in 10-bit address mode (the I2C sends the target address second byte every time it sends the target address first byte)
-

### 16.1.1 Introduction to the I2C Module

The I2C module supports any target or controller I2C-compatible device. [Figure 16-1](#) shows an example of multiple I2C serial ports connected for a two-way transfer from one device to another device.



**Figure 16-1. Multiple I2C Modules Connection Diagram**

### 16.1.2 Functional Overview

The I2C module is a serial bus that supports multiple Controller devices. In multi Controller mode, one or more devices can be connected to the same bus and are capable of controlling the bus. Each I2C device on the bus is recognized by a unique address and can operate as either a transmitter or a receiver, depending on the function of the device. In addition to being a transmitter or receiver, a device connected to the I2C bus can also be considered a Controller or a Target when performing data transfers.

#### Note

A Controller device is the device that initiates the data transfer on a bus and generates the clock signal that permits the transfer. During the transmission, any device addressed by the Controller is considered the Target.

Data is communicated to devices interfacing to the I2C module using the serial data pin (SDA) and the serial clock pin (SCL) as shown in [Figure 16-2](#). These two wires carry information between the device and the other devices connected to the I2C bus. Both SDA and SCL pins on the device are bidirectional. They must be connected to a positive supply voltage through a pull-up resistor. When the bus is free, both pins are high. The driver of these two pins has an open-drain configuration to perform the wired-AND function.

The device has a special mode that can be entered to ignore a NACK generated from non-compliant I2C devices that are incapable of generating an ACK.

The I2C module consists of the following Controller blocks:

- A serial Interface: one data pin (SDA) and one clock pin (SCL)
- The device register interface
  - Data registers to temporarily hold received data and transmitted data traveling between the SDA pin and the CPU or the DMA
  - Control and status registers
- A prescaler to divide down the input clock that is driven to the I2C module
- A peripheral bus interface to enable the CPU and DMA to access the I2C module registers
- An arbitrator to handle arbitration between the I2C module (when configured as a Controller) and another Controller
- Interrupt generation logic (interrupts can be sent to the CPU)
- A clock synchronizer that synchronizes the I2C input clock (from the system module) and the clock on the SCL pin, and synchronizes data transfers with controllers of different clock speeds.
- A noise filter on each of the two serial pins
- DMA event generation logic that synchronizes data reception and data transmission in the I2C module for DMA transmission

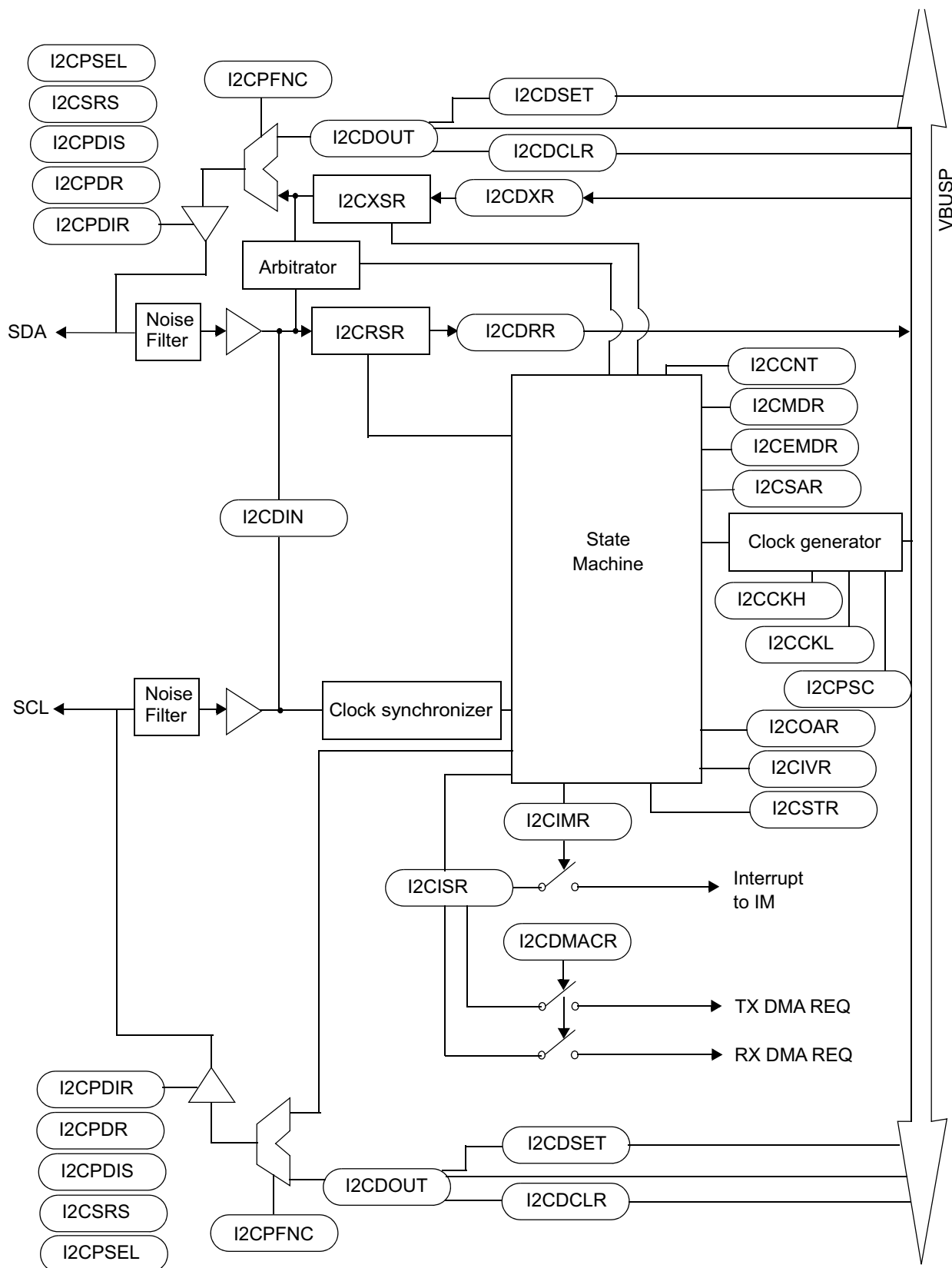
In [Figure 16-2](#), the CPU or the DMA writes data for transmission to I2CDXR and reads received data from I2CDRR. When the I2C module is configured as a transmitter, data written to I2CDXR is copied to I2CXS

and shifted out one bit at a time. When the I2C module is configured as a receiver, received data is shifted into I2CRSR and then copied to I2CDRR.

When the I2C function is not needed, the pins may be controlled as general-purpose input/output (GPIO) pins. The I/O structure of each pin includes:

- programmable slew rate control of the outputs
- open drain mode
- programmable pull enable/disable on the input
- programmable pull up/pull down function on the input

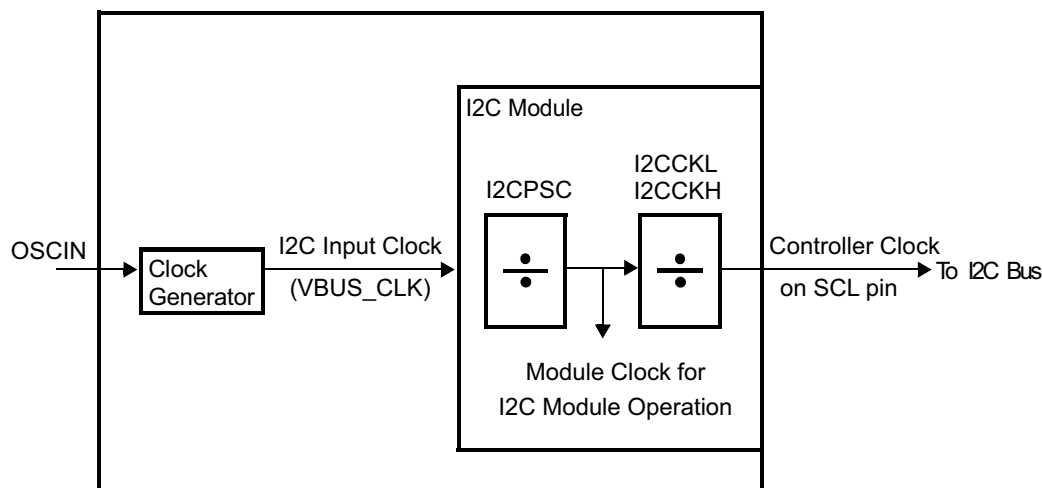




### Figure 16-2. Simple I2C Block Diagram

### 16.1.3 Clock Generation

As shown in Figure 16-3, the I2C module uses the input clock generated from the device clock generator to generate the module clock and Controller clock. The I2C input clock is the device peripheral clock (VBUS\_CLK). The clock is then divided twice more inside the I2C module to produce the module clock and the Controller clock.



**Figure 16-3. Clocking Diagram for the I2C Module**

The module clock determines the frequency at which the I2C module operates. A programmable prescaler in the I2C module divides down the input clock to produce the module clock. To specify the divide-down value, initialize the I2CPSC field of the prescaler register, I2CPSC. The resulting frequency is:

$$\text{ModuleClockFrequency} = \frac{\text{I2CInputClockFrequency}}{(\text{ICPSC} + 1)} \quad (5)$$

The module clock frequency must be between 6.7MHz and 13.3MHz. The prescaler can only be initialized while the I2C module is in the reset state (IRS = 0 in I2CMDR). The prescaled frequency takes effect only when IRS is changed to 1. Changing the I2CPSC value while IRS = 1 has no effect.

The Controller clock appears on the SCL pin when the I2C module is configured to be a Controller on the I2C bus. This clock controls the timing of the communication between the I2C module and a secondary. As shown in Figure 16-3, a second clock divider in the I2C module divides down the module clock to produce the Controller clock. The clock divider uses the I2CCKL to divide down the low portion of the module clock signal and uses the I2CCKH to divide down the high portion of the module clock signal.

The resulting frequency is:

$$\text{ControllerClockFrequency} = \frac{\text{ModuleClockFrequency}}{(\text{ICCLKL} + d) + (\text{ICCLKH} + d)} \quad (6)$$

$$\text{ControllerClockFrequency} = \frac{\text{I2CInputClockFrequency}}{(\text{ICPSC} + 1)((\text{ICCLKL} + d) + (\text{ICCLKH} + d))} \quad (7)$$

where  $d$  depends on the value of I2CPSC:

I2CPSC	d
0	7
1	6

I2CPSC	d
Greater than 1	5

### Note

The Controller clock frequency defined above does not include rise/fall time and latency of the synchronizer inside the module. The actual transfer rate will be slower than the value calculated from the formula above. Also, due to the nature of SCL synchronization, the SCL clock period could change if SCL synchronization is taking place.

## 16.2 I2C Module Operation

The following section discusses how the I2C module operates.

### 16.2.1 Input and Output Voltage Levels

One clock pulse is generated by the Controller device for each data bit transferred. Because of a variety of different technology devices that can be connected to the I2C-bus, the levels of logic 0 (low) and logic 1 (high) are not fixed and depend on the associated level of  $V_{CCIO}$ . For details, see the device specific data sheet.

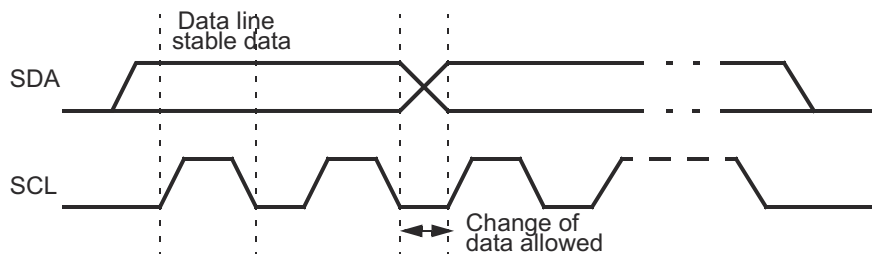
### 16.2.2 I2C Module Reset Conditions

The I2C module can be reset in the following two ways:

- Through the global peripheral reset. A device reset causes a global peripheral reset.
- By clearing the  $\overline{IRS}$  bit in the I2C mode register (I2CMDR). When the global peripheral reset is removed, the  $\overline{IRS}$  bit is cleared to 0, keeping the I2C module in the reset state.

### 16.2.3 I2C Module Data Validity

The data on the SDA must be stable during the high period of the clock. See Figure 16-4. The high and low state of the data line, the SDA, can only change when the clock signal on the serial clock line (SCL) is low.

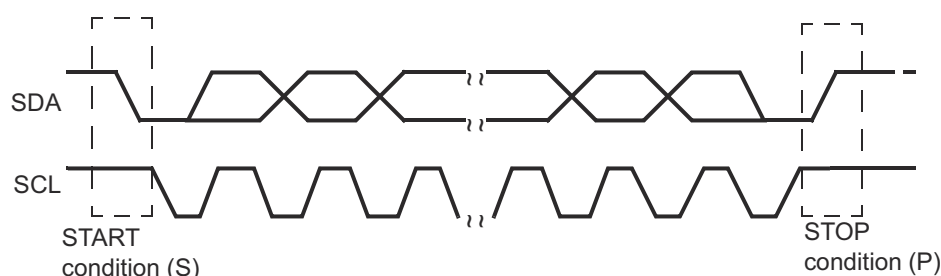


**Figure 16-4. Bit Transfer on the I2C Bus**

### 16.2.4 I2C Module Start and Stop Conditions

START and STOP conditions are generated by a primary I2C module.

- The START condition is defined as a high-to-low transition on the SDA line while SCL is high. A primary drives this condition to indicate the start of data transfer. The bus is considered to be busy after the START condition, and the bus busy bit (BB) in I2CSR is set to 1.
- The STOP condition is defined as a low-to-high transition on the SDA line while SCL is high. A primary drives this condition to indicate the end of data transfer. The bus is considered to be free after the STOP condition, therefore the BB bit in I2CSR is cleared to 0.

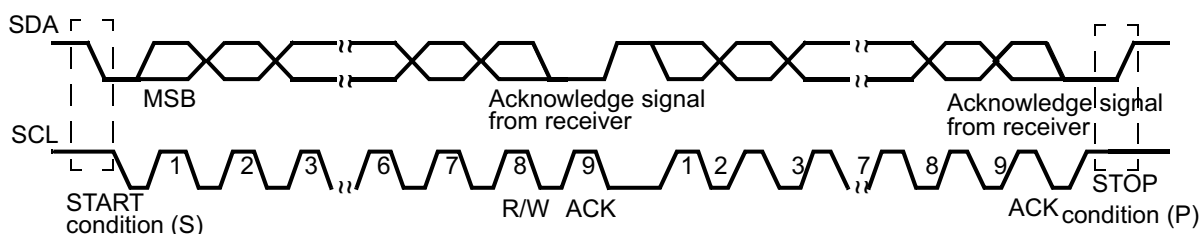


**Figure 16-5. I2C Module START and STOP Conditions**

For the I2C module to start a data transfer with a START condition, the primary mode bit (MST) and the START condition bit (STT) in the I2CMDR must both be set to 1. For the I2C module to end a data transfer with a STOP condition, the STOP condition bit (STP) must be set to 1. When the BB bit is set to 1 and the STT bit is set to 1, a repeated START condition is generated.

### 16.2.5 Serial Data Formats

The I2C module operates in byte data format. Each message put on the SDA line is 2 to 8-bits long. The number of messages that can be transmitted or received is unrestricted. The data is transferred with the most significant bit (MSB) first (Figure 16-6). Each message is followed by an acknowledge bit from the I2C if it is in receiver mode. The I2C module does not support little endian systems.



**Figure 16-6. I2C Module Data Transfer**

The first byte after a START condition (S) always consists of 8 bits that comprise either a 7-bit address plus the R/  $\overline{W}$  bit, or 8 data bits. The eighth bit, R/  $\overline{W}$ , in the first byte determines the direction of the data. When the R/  $\overline{W}$  bit is 0, the Controller writes (transmits) data to a selected Target device; when the R/  $\overline{W}$  bit is 1, the Controller reads (receives) data from the Target device. In acknowledge mode, an extra bit dedicated for the acknowledgment (ACK) bit is inserted after each message.

The I2C module supports the following formats:

- 7-bit addressing format (Figure 16-7)
- 10-bit addressing format (Figure 16-8)
- 7-bit/10-bit addressing format with repeated START condition (Figure 16-9)
- Free-data format (Figure 16-10)

### 16.2.5.1 7-Bit Addressing Format

In the 7-bit addressing format (Figure 16-7), the first byte after the START condition consists of a 7-bit secondary address followed by the R/  $\overline{W}$  bit (in the LSB). The R/  $\overline{W}$  bit determines the direction of the data transfer:

- R/  $\overline{W}$  = 0: The primary writes (transmits) data to the addressed secondary.
- R/  $\overline{W}$  = 1: The primary reads (receives) data from the secondary.

An extra clock cycle dedicated for acknowledgement (ACK) is inserted after each byte. If the ACK is inserted by the secondary after the first byte from the primary, it is followed by n bits of data from the transmitter (primary or secondary, depending on the R/  $\overline{W}$  bit). The device I2C allows n to be a number between 2 to 8, programmable by the bit count (BC) field of I2CMR. After the data bits have been transferred, the receiver inserts an ACK bit.

To select the 7-bit addressing format, write 0 to the expanded address enable (XA) bit of I2CMR and make sure the free data format mode is off (FDF = 0 in I2CMR).

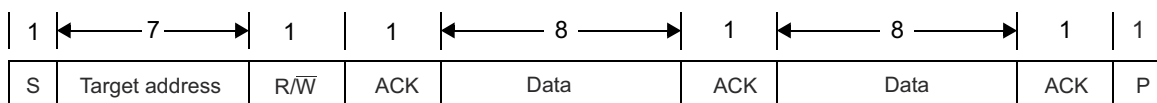


Figure 16-7. I2C Module 7-Bit Addressing Format

### 16.2.5.2 10-Bit Addressing Format

The 10-bit addressing format is similar to the 7-bit addressing format, but the primary sends the secondary address in two separate byte transfers. In the 10-bit addressing format (Figure 16-8), the first byte is 11110b, the two MSBs of the 10-bit secondary address, and the R/  $\overline{W}$  bit. The ACK bit is inserted after each byte.

The second byte is the remaining 8 bits of the 10-bit secondary address. The secondary must send an acknowledgment after each of the two byte transfers. Once the primary has written the second byte to the secondary, the primary can either write data or use repeated a START condition to change the data direction.

To select the 10-bit addressing format, write 1 to the expanded address enable (XA) bit of I2CMR and make sure the free data format mode is off (FDF = 0 in I2CMR).

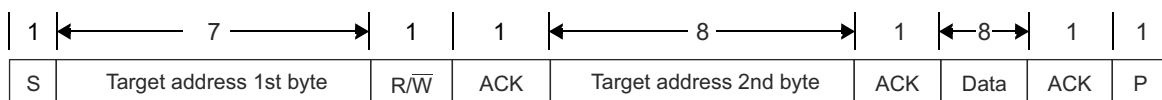


Figure 16-8. I2C Module 10-bit Addressing Format

### 16.2.5.3 Using the Repeated START Condition

At the end of each byte, the primary can drive another START condition (Figure 16-9). Using this capability, a primary can transmit/receive any number of data bytes before generating a STOP condition. The length of a data byte can be from 2 to 8 bits. The repeated START condition can be used with the 7-bit addressing, 10-bit addressing, or the free data formats.

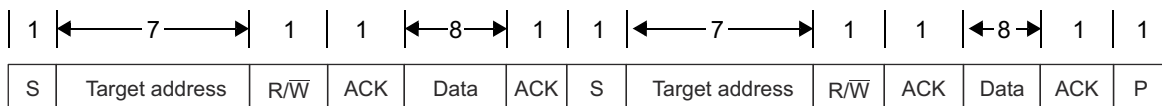
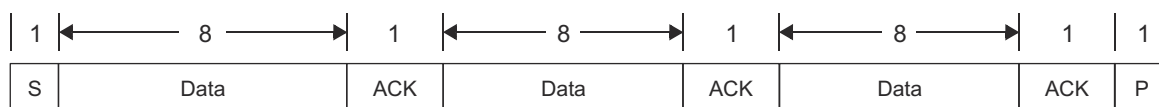


Figure 16-9. I2C Module 7-Bit Addressing Format with Repeated START

### 16.2.5.4 Free Data Format

In this format (Figure 16-10), the first byte after a START condition is a data byte. The ACK bit is inserted after each byte, followed by another 8 bits of data. No address or data direction bit is sent. Therefore, the transmitter and receiver must both support the free data format. The direction of data transmission (transmit or receive) remains constant throughout the transfer.

To select the free data format, write a 1 to the free data format (FDF) bit of the I2CMDR. The free data format is not supported in the digital loop back mode.



**Figure 16-10. I2C Module in Free Data Format**

### 16.2.6 NACK Bit Generation

When the I2C module is a receiver (Controller or Target), it can acknowledge or ignore bits sent by the transmitter. To ignore any new bits, the I2C module must send a no-acknowledge (NACK) bit during the acknowledge cycle on the bus. Table 16-1 summarizes the various ways a NACK can be generated.

**Table 16-1. Ways to Generate a NACK Bit**

I2C Module Condition	Basic NACK Bit Generation Options	Additional Option
Target receiver mode	Disable data transfers (STT = 0) Allow an overrun condition (RSFULL = 1) Reset the module (IRS = 0)	Set the NACKMOD bit before the rising edge of the last data bit you intend to receive.
Controller receiver mode and repeat mode (RM = 1)	Generate a STOP condition (STP = 1) Reset the module (IRS = 0)	Set the NACKMOD bit before the rising edge of the last data bit you intend to receive.
Controller receiver mode with non-repeat mode (RM = 0)	If STP = 1, allow the internal data counter to count down to 0 and thus force a STOP condition. If STP = 0, make STP = 1 to generate a STOP condition. Reset the module (IRS = 0)	Set the NACKMOD bit before the rising edge of the last data bit you intend to receive.

In some applications, the Target cannot generate the ACK signal. If the IGNACK bit is set in the I2CEMDR register, the resulting NACK will be ignored and the I2C block will continue the data transfer.

## 16.3 I2C Operation Modes

### 16.3.1 Controller Transmitter Mode

All primaries begin in this mode. The I2C module is a Controller and transmits control information and data to a Target. In this mode, data assembled in any of the addressing formats shown in [Figure 16-7](#), [Figure 16-8](#), or [Figure 16-9](#) is shifted out onto the SDA pin and synchronized with the self-generated clock pulses on the SCL pin. The clock pulses are inhibited and the SCL pin is held low when the intervention of the device is required ( $\overline{\text{XSMT}} = 0$ ) after a byte has been transmitted.

#### Note

If the I2C is configured for two simultaneous Controller transmissions, wait until the MST and BB have been reset before performing the second Controller transmission.

Failure to wait for the MST and BB to reset will prevent the start condition on the second transfer from being issued and the bus BB will not be set. Typically the end of the first transfer is handled by polling BB. However, the MST bit is not reset at the same instant as the BB bit. As a result, when the second Controller transmission is initiated before the resetting of the MST, the MST bit for the second transfer is reset. This prevents the I2C from recognizing itself as the Controller, thus failing to occupy the bus.

### 16.3.2 Controller Receiver Mode

In this mode, the I2C module is a Controller and receives data from a Target. This mode can only be entered from the Controller transmitter mode (the I2C module must first transmit a command to the Target). In any of the addressing formats shown in [Figure 16-7](#), [Figure 16-8](#), or [Figure 16-9](#), the Controller receiver mode is entered after the Target address byte and the R/  $\overline{\text{W}}$  bit have been transmitted (if the R/  $\overline{\text{W}}$  bit is 1). Serial data bits received on the SDA pin are shifted in with the self-generated clock pulses on the SCL pin. The clock pulses are inhibited and the SCL is held low when the intervention of the device is required (RSFULL = 1) after a byte has been received. At the end of the transfer, the Controller-receiver signals the end of data to the Target-transmitter by not generating an acknowledge on the last byte that was clocked out of the Target. The Target-transmitter then releases the data line allowing the Controller-receiver to generate a STOP condition or a repeated START condition.

In many applications, the size of the message is in the initial bytes of the message itself. Since the size of the message is not known to the Controller before the transmission/reception starts, the Controller must use the repeat mode to force the stop condition when the reception is completed. The repeat mode is enabled by setting the RM bit to 1. Due to the double buffer implementation on the receive side, the Controller must generate the stop condition (STP = 1) after reading the (message size - 1)<sup>th</sup> data.

### 16.3.3 Target Transmitter Mode

In this mode, the I2C module is a Target and transmits data to a Controller. This mode can only be entered from the Target receiver mode (The I2C module must first receive a command from the Controller). In any of the addressing formats shown in [Figure 16-7](#), [Figure 16-8](#), or [Figure 16-9](#), the Target transmitter mode is entered if the Target address byte is the same as its own address and the R/  $\overline{\text{W}}$  bit has been transmitted (if the R/  $\overline{\text{W}}$  bit is set to 1). The Target transmitter shifts the serial data out on the SDA pin with the clock pulses that are generated by the Controller device. The Target device does not generate the clock, but it can hold the SCL pin low when intervention of the device is required ( $\overline{\text{XSMT}} = 0$ ) after a byte has been transmitted.

### 16.3.4 Target Receiver Mode

In this mode, the I2C module is a Target and receives data from a Controller. All Target begin in this mode. Serial data bits received on the SDA pin are shifted in with the clock pulses that are generated by the Controller device. The Target device does not generate the clock, but it can hold the SCL pin low while intervention of the device is required (RSFULL = 1) after a byte has been received.

### **16.3.5 Free Run Mode**

The I2C module can be placed in free run mode when the FREE bit (I2CMR.14) is set to 1. This bit is primarily used on an emulator when encountering a break point while debugging software. When the FREE bit is set to 0, the I2C responds differently depending on whether the SCL is high or low. If the SCL is low, the I2C stops immediately and keeps driving the SCL low whether the I2C is the Controller transmitter or receiver. If the SCL is high, the I2C waits until the SCL becomes a low and then stops. If the I2C is a Target, it stops when the transmission/reception completes.

### **16.3.6 Ignore NACK Mode**

The I2C module can be placed in the ignore NACK mode by setting the IGNACK bit in the I2CEMR register. This mode allows an I2C module that is configured as a Controller transmitter to ignore a NACK from a Target device that is not capable of generating a proper ACK signal.



## 16.4 I2C Module Integrity

The following section discusses how the I2C module maintains priorities and order among signals and commands.

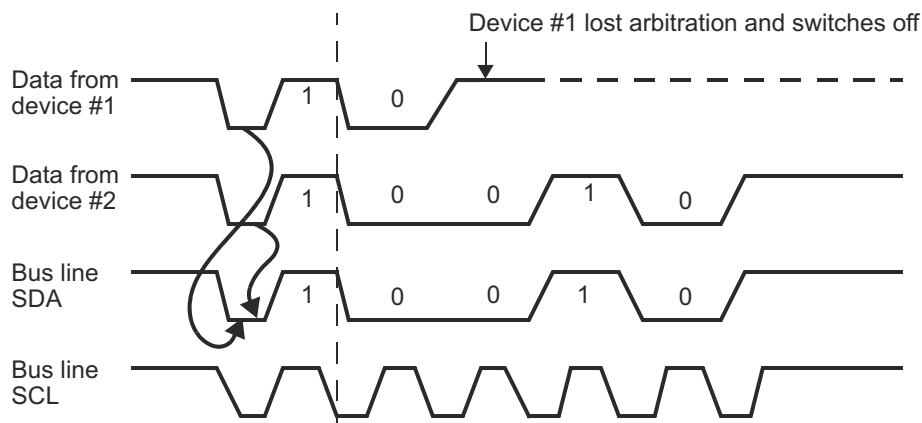
### 16.4.1 Arbitration

If two or more Controller transmitters simultaneously start a transmission on the same bus, an arbitration procedure is invoked. [Figure 16-11](#) illustrates the arbitration procedure between two devices. The arbitration procedure uses the data presented on the SDA bus by the competing transmitters. The first Controller transmitter that generates a high is overruled by the other Controller that generates a low. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. The Controller transmitter that loses the arbitration switches to the Target receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration-lost interrupt. The data transmitted by the other Controller module is salvaged, and the I2C continues to receive data from the Controller module. Should two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

If, during a serial transfer, the arbitration procedure is still in progress when a repeated START condition or STOP condition is transmitted to I2C bus, the Controller transmitters involved must send the repeated START condition or STOP condition at the same position in the format frame. In other words, arbitration is not allowed between:

- A repeated START condition and a data bit
- A STOP condition and a data bit
- A repeated START condition and a STOP condition

Secondaries are not involved in the arbitration procedure.



**Figure 16-11. Arbitration Procedure Between Two Controller Transmitters**

### 16.4.2 I2C Clock Generation and Synchronization

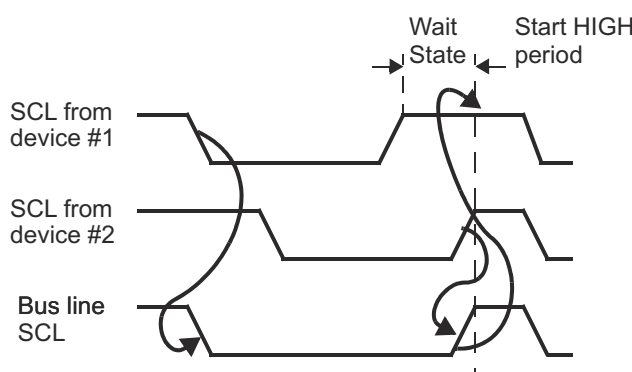
Under normal conditions only one Controller device generates the clock signal; the SCL. During the arbitration procedure, however, there are two or more Controller devices and the clock must be synchronized so that the data output can be compared. [Figure 16-12](#) illustrates clock synchronization. The wired-AND property of the SCL line means that a device that first generates a low period on the SCL overrules the other devices. At this high-to-low transition, the clock generators of the other devices are forced to start their own low period. The SCL line is held low by the device with the longest low period. The other devices that finish their low periods must wait for the SCL line to be released before starting their high periods. A synchronized signal on the SCL is obtained where the slowest device determines the length of the low period and the fastest device determines the length of the high period.

If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the wait state. In this way, a Target slows down a fast Controller and the slow device creates enough time to store a received byte or to prepare a byte to be transmitted.

#### Note

##### I2C Protocol Fault

The following conditions violate the clock spec as defined in the Philips I<sup>2</sup>C bus specification, v2.1 (*The I<sup>2</sup>C Specification*, Philips document number 9398 393 40011), and will result in an I2C protocol fault: I2CCLKH = 2 I2CCLKL = 2I2CPSC = 2. This will cause the SDA data transition to occur while the SCL is high.



**Figure 16-12. Synchronization of Two I2C Clock Generators During Arbitration**

### 16.4.3 Prescaler

The I2C module is operated by the module clock. This clock is generated by way of the I2C prescaler block. The prescaler block consists of a 8-bit register, I2CPSC, used for dividing down the device peripheral clock (VBUS\_CLK) to obtain a module clock between 6.7 MHz and 13.3 MHz.

### 16.4.4 Noise Filter

The noise filter is used to suppress any noises that are 50ns or less. It is designed to suppress noise with one module clock, assuming the lower and upper limits of the module clock are 6.7MHz and 13.3MHz, respectively.

## 16.5 Operational Information

The following section provides specific information about how the I2C module operates.

### 16.5.1 I2C Module Interrupts

The I2C module generates seven types of interrupts. These seven interrupts are accompanied with seven interrupt mask bits in the interrupt mask register (I2CIMR) and with seven interrupt flag bits in the status register (I2CSR).

#### 16.5.1.1 I2C Interrupt Requests

The I2C module generates the interrupt requests described below. All requests are multiplexed through an arbiter into a single I2C interrupt request to the CPU. Each interrupt request has a flag bit and an enable bit. Interrupts must be enabled prior to the occurrence of the expected interrupt condition. When one of the specified events occurs, the flag bit is set. If the corresponding enable bit is 0, the interrupt request is blocked. If the enable bit is 1, the interrupt request is forwarded to the CPU as an I2C interrupt request. As an alternative, the CPU can poll all of the bits shown in [Table 16-2](#).

**Table 16-2. Interrupt Requests Generated by I2C Module**

Flag	Name	Generated
AL	Arbitration-lost interrupt	Generated when the I2C module has lost an arbitration contest with another Controller-transmitter
NACK	No-acknowledge interrupt	Generated when the Controller I2C does not receive an acknowledge from the receiver
ARDY	Register-access-ready interrupt	Generated when the previously programmed address, data and command have been performed and the status bits have been updated. The interrupt is used to notify the device that the I2C registers are ready to be accessed.
RXRDY	Receive-data-ready interrupt	Generated when the received data in the receive-shift register (I2CSR) has been copied into the data receive register (I2CDRR). The RXRDY bit can also be polled by the device to determine when to read the received data in the I2CDRR.
TXRDY	Transmit-data-ready interrupt	Generated when the transmitted data has been copied from the data transmit register (I2CDXR) into the transmit-shift register (I2CXSR). The TXRDY bit can also be polled by the device to determine when to write the next data into I2CDXR.
SCD	Stop-condition-detect interrupt	Generated when a STOP condition has been detected.
AAS	Address-as-Target interrupt	Generated when the I2C has recognized its own Target address or an address of all zeroes.

### 16.5.2 DMA Controller Events

The I2C module has two events that use the DMA controller to synchronously read received data (I2CREVNT) from I2CDRR, and synchronously write data (I2CWEVNT) to the transmit buffer, I2CDXR. The read and write events have the same timing as I2CRRDY (I2CRINT) and I2CXRDY (I2CXINT), respectively.

The CPU or the DMA controller reads the received data from I2CDRR and writes the data to be transmitted to I2CDXR. The RXRDY bit is automatically cleared when the DMA controller reads the I2CDRR register, and the TXRDY bit is automatically cleared when the DMA controller writes to the I2CDXR register.

Data written to I2CDXR is copied to I2CXSR and shifted out from the SDA pin when the I2C module is configured as a transmitter. When the I2C module is configured as a receiver, received data is shifted into ICRSR and copied to I2CDRR, which can be read by the CPU or the DMA controller.

A transmit event (I2CWEVNT) is generated after a START condition in Controller transmitter mode. This ensures that the DMA gets an event even if no Target returns an ACK to the Target address following the START condition.

---

**Note****Unexpected DMA transmit and receive event**

An unexpected DMA transmit event (ICXEVT) and a DMA receive event (ICXRDY) are generated in 10-bit, Controller transmit, repeat mode. This event occurs soon after the start condition but before the first bit of the address is transmitted. In this event, no DMA activity should be initiated without the Target ACK being received.

---

**16.5.3 I2C Enable/Disable**

The I2C module can be enabled or disabled with the I2C reset enable bit (IRS) in the I2C module register (I2CMDR). This occurs in one of two ways:

- Write 0 to the I2C reset bit (IRS) in I2CMDR. All status bits are forced to the default values and the I2C mode remains disabled until IRS is changed to 1. The SDA and SCL pins are in the high impedance state.
- Initiate a device reset by driving the  $\overline{\text{PORRST}}$  pin low. The entire device is reset and is held in the reset state until the pin is released and is driven high. When  $\overline{\text{PORRST}}$  is released, all I2C module registers are reset to their default values. The IRS bit is forced to 0, which resets the I2C module. The I2C module stays in the reset state until a 1 is written to the IRS bit.

IRS must be 0 while the I2C module is being configured. Forcing IRS to 0 can be used to save power and also clear error conditions.

**16.5.4 General Purpose I/O**

Both of the I2C pins can be programmed to be general-purpose I/O pins via the I2C pin control registers (I2CPFNC, I2CDIR, I2CDOUT, and I2CDIN).

When the I2C module is not used, the I2C pins may be programmed to be either general purpose input or general-purpose output pins. This function is controlled in the I2CDIR and I2CPFNC registers. Note that each pin can be programmed to be either an I2C pin or a GIO pin.

If the I2C function is to be used, the application software must ensure that each pin is configured as an I2C pin and not a GIO pin, or else unexpected behavior may result.

### **16.5.5 Pull Up/Pull Down Function**

I2C module pins can have either an active pull up or active pull down that makes it possible to leave the pins unconnected externally. The pins can be programmed to have the active pull function enabled or disabled by writing to the corresponding bit in the I2CPDIS register. Please see the device-specific data sheet for the default internal pull (pull-up, pull-down or no pull) on the pins.

The pull on the pins is programmable to a setting other than the default internal pull as specified in the data sheet. The pins can be programmed to have either an active pull up or an active pull down function by writing to the corresponding bit in I2CPSEL register. The pull up/pull down function is active on the pin only when the pull enabled is programmed in the I2CPDIS register.

The pull up/pull down functions are deactivated when a bidirectional pin is configured as an output. At system reset, the pull up function of all the pins is enabled. Please see the device-specific data sheet for the current supplied by the pull up/pull down.

### **16.5.6 Open Drain Function**

The I2C pins can be programmed to include an open drain function when they are configured as output pins. This is done by writing to the corresponding bit of the I2CPDR register. When the open drain function is enabled, a low value (0) written to the data output register forces the pin to a low output voltage ( $V_{OL}$  or lower), whereas a high value (1) written to the data output register forces the pin to a high-impedance state. The open drain function is disabled when the pin is configured as an input pin.

## 16.6 MSS\_I2C Registers

Table 16-3 lists the memory-mapped registers for the MSS\_I2C registers. All register offset addresses not listed in Table 16-3 should be considered as reserved locations and the register contents should not be modified.

**Table 16-3. MSS\_I2C Registers**

Offset	Acronym	Register Name	Section
0h	ICOAR	ICOAR	<a href="#">Go</a>
4h	ICIMR	ICIMR	<a href="#">Go</a>
8h	ICSTR	ICSTR	<a href="#">Go</a>
Ch	ICCLKL	ICCLKL	<a href="#">Go</a>
10h	ICCLKH	ICCLKH	<a href="#">Go</a>
14h	ICCNT	ICCNT	<a href="#">Go</a>
18h	ICDRR	ICDRR	<a href="#">Go</a>
1Ch	ICSAR	ICSAR	<a href="#">Go</a>
20h	ICDXR	ICDXR	<a href="#">Go</a>
24h	ICMDR	ICMDR	<a href="#">Go</a>
28h	ICIVR	ICIVR	<a href="#">Go</a>
2Ch	ICEMDR	ICEMDR	<a href="#">Go</a>
30h	ICPSC	ICPSC	<a href="#">Go</a>
34h	ICPID1	ICPID1	<a href="#">Go</a>
38h	ICPID2	ICPID2	<a href="#">Go</a>
3Ch	ICDMAC	ICDMAC	<a href="#">Go</a>
40h	I2C_RESERVED1	I2C_RESERVED1	<a href="#">Go</a>
44h	I2C_RESERVED2	I2C_RESERVED2	<a href="#">Go</a>
48h	ICPFUNC	ICPFUNC	<a href="#">Go</a>
4Ch	ICPDIR	ICPDIR	<a href="#">Go</a>
50h	ICPDIN	ICPDIN	<a href="#">Go</a>
54h	ICPDOUT	ICPDOUT	<a href="#">Go</a>
58h	ICPDSET	ICPDSET	<a href="#">Go</a>
5Ch	ICPDCLR	ICPDCLR	<a href="#">Go</a>
60h	ICPDRV	ICPDRV	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. Table 16-4 shows the codes that are used for access types in this section.

**Table 16-4. MSS\_I2C Access Type Codes**

Access Type	Code	Description
<b>Read Type</b>		
R	R	Read
<b>Write Type</b>		
W	W	Write
<b>Reset or Default Value</b>		
-n		Value after reset or the default value

### 16.6.1 ICOAR Register (Offset = 0h) [Reset = 00000000h]

ICOAR is shown in [Table 16-5](#).

Return to the [Summary Table](#).

I2C Own Address register

**Table 16-5. ICOAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	NU	R/W	0h	Reserved
9-0	A9_A0	R/W	0h	Own address. Use in both 7- and 10-bit address mode. Note that user can program the I2C own address to any value as long as it does not conflict with other components in the system.

## 16.6.2 ICIMR Register (Offset = 4h) [Reset = 00000000h]

ICIMR is shown in [Table 16-6](#).

Return to the [Summary Table](#).

I2C Interrupt Mask/Status register

**Table 16-6. ICIMR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	NU	R/W	0h	Reserved
6	AAS	R/W	0h	Address As Slave interrupt mask bit. Setting a "1" to this bit unmask the Address As Slave interrupt. Setting a "0" to this bit masks the Address As Slave interrupt.
5	SCD	R/W	0h	Stop Condition Detection mask bit. Setting a "1" to this bit unmask the Stop Condition Detection interrupt. Setting a "0" to this bit masks the Stop Condition Detection interrupt.
4	ICXRDY	R/W	0h	Transmit Data Ready interrupt mask bit. Setting a "1" to this bit unmask the Transmit Data Ready interrupt. Setting a "0" to this bit masks the Transmit Data Ready interrupt.
3	ICRRDY	R/W	0h	Receive Data Ready interrupt mask bit. Setting a "1" to this bit unmask the Receive Data Ready interrupt. Setting a "0" to this bit masks the Receive Data Ready interrupt.
2	ARDY	R/W	0h	Register access ready interrupt mask bit. Setting a "1" to this bit unmask the Register access ready interrupt. Setting a "0" to this bit masks the Register access ready interrupt.
1	NACK	R/W	0h	No Acknowledgement interrupt mask bit. Setting a "1" to this bit unmask the No Acknowledgement interrupt. Setting a "0" to this bit masks the No Acknowledgement interrupt.
0	AL	R/W	0h	Arbitration Lost interrupt mask bit. Setting a "1" to this bit unmask the Arbitration Lost interrupt. Setting a "0" to this bit masks the Arbitration Lost interrupt.



### 16.6.3 ICSTR Register (Offset = 8h) [Reset = 00000000h]

ICSTR is shown in [Table 16-7](#).

Return to the [Summary Table](#).

I2C Interrupt Status register

**Table 16-7. ICSTR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-15	NU2	R/W	0h	Reserved
14	SDIR	R/W	0h	Slave Direction. This bit is clear to '0' indicating the I2C is a master transmitter/receiver or a slave receiver. This bit is also clear by STOP condition or START condition. It is set to '1' when the I2C slave is a transmitter. In DLB mode (which the configuration should be master-transmitter slave-receiver) this bit is clear to '0'. Writing a "1" to this bit to clear it.
13	NACKSNT	R/W	0h	A No Acknowledge is sent due to NACKMOD is set to a "1". NACKSNT = 0: A No Acknowledge is not sent. NACKSNT = 1: A No Acknowledge is sent. Writing a "1" to this bit to clear it.
12	BB	R/W	0h	Bus Busy. This bit indicates the state of the serial bus. BB= 0: The bus is free. BB= 1: The bus is occupied. On reception of a "start" condition the device sets BB to 1. This bit is also set if the I2C detects SCL low state. BB is clear to 0 after reception of a "stop" condition. BB is kept to "0" regardless SCL state when the I2C is in reset (IRS_ = 0). If the IRS_ is set to "1" during transaction between other I2C devices the BB bit is set at the first falling edge of SCL or START condition. - (RW )
11	RSFULL	R/W	0h	Receive shift full. This bit indicates whether the receiver has experienced overrun. Overrun occurs when the receive shift register (ICRSR) is full and ICDRR has not been read since the ICRSR-to-ICDRR transfer. The FSM is holding for ICDRR read access. RSFULL is clear when reading the ICDRR. RSFULL is set to "1" when the I2C has recognized an overrun. The contents of ICDRR are NOT lost in this case. In repeat mode since double buffer (ICRSR and ICDRR) behaves like a single buffer RSFULL is set to "1" every time the data is received. RSFULL is clear as a result of reading the ICDRR. - (RW )
10	XSMT	R/W	0h	Transmit shift empty not. This bit indicates whether the transmitter has experienced underflow. Underflow occurs when the transmit shift register (ICXSR) is empty and ICDEXR has not been loaded. The FSM is holding for ICDEXR write access. XSMT_ is cleared when underflow has occurred. XSMT_ is set to "1" as a result of writing to ICDEXR. In repeat mode if the I2C in master transmitter mode is holding transfer with XSMT_ = 0 (i.e. waiting for further action) and the STT or STP bit is set XSMT_ is set to "1" by hardware.

**Table 16-7. ICSTR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9	AAS	R/W	0h	Address As Slave. This bit is set to 1 by the device when it has recognized its own slave address or an address of all (8) zeros. The AAS bit is reset by stop condition or detection of any address byte that does not match ICOAR. - (RW )
8	AD0	R/W	0h	Address Zero Status: This bit is set to 1 by device if it detects the address of all (8) zeros (i.e. general call). The AD0 bit is reset to 0 (default value) when a "start" or "stop" condition is detected. - (RW )
7-6	NU1	R/W	0h	Reserved
5	SCD	R/W	0h	Stop Condition Detection bit SCD is set when the I2C sends or receives STOP condition. This bit is cleared by reading ICIVR (as 110) or writing '1' to itself.
4	ICXRDY	R/W	0h	Transmit Data Ready interrupt flag bit. ICXRDY is set to "1" is generated when the transmitted data has been copied from ICDXR to the transmit-shift register (ICXSR). ICXRDY is clear to "0" when the ICDXR is written. This bit can also be polled by the CPU to write a new transmitted data into the ICDXR. Write '1' to this bit will set it and DXR Write will clear it.
3	ICRRDY	R/W	0h	Receive Data Ready interrupt flag bit. ICRRDY is set to "1" when the received data has been copied from ICRSR into the ICDRR. ICRRDY is cleared to "0" when the ICDRR is read. This bit can also be polled by the CPU to read the received data in the ICDRR. Write '1' or DRR Read will clear it.
2	ARDY	R/W	0h	Register-access-ready interrupt flag bit. ARDY is generated by the hardware if the I2C is in the master mode when the previously programmed data and command has been performed and status bit has been updated. This flag is used by the CPU to let it knows that the I2C registers are ready to be accessed again. When RM=0 ARDY is set when the internal data count is passed 0 if STP register bit has not been set. When RM=1 ARDY is set at each byte end. If the I2C is in FDF mode(FDF=1) ARDY is set just after Start condition. This bit is automatically cleared by hardware when writing data to ICDXR in transmit mode reading data from ICDRR in receive mode or setting STT or STP bit. Write '1' will clear it.
1	NACK	R/W	0h	No-Acknowledgement interrupt flag bit. The No Acknowledge flag bit is set when the hardware in "master" mode detects no acknowledge has been received. This bit is NOT set by no-acknowledgement after Start byte Write '1' or Read the ICIVR (as 010) will clear it.
0	AL	R/W	0h	Arbitration-Lost interrupt flag bit. The Arbitration Lost flag bit is set to 1 when the device in the "master" mode senses it has lost an arbitration when two or more transmitters start a transmission almost simultaneously or when the I2C attempts to start a transfer while BB (bus busy) is 1. When this is set to 1 due to arbitration lost the MST/STT/STP bits are clear the I2C becomes a slave. Write '1' or Read the ICIVR (as 001) will clear it.

### 16.6.4 ICCLKL Register (Offset = Ch) [Reset = 00000000h]

ICCLKL is shown in [Table 16-8](#).

Return to the [Summary Table](#).

I2C Clock Divider Low register

**Table 16-8. ICCLKL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved
15-0	ICCL15_ICCL0	R/W	0h	Low time I 2 C SCL Clock Division Factor. They are used to divide down the master clock to create the SCL low time transition frequency. This register must be configured while the I2C is still in reset (IRS_=0).

### 16.6.5 ICCLKH Register (Offset = 10h) [Reset = 00000000h]

ICCLKH is shown in [Table 16-9](#).

Return to the [Summary Table](#).

I2C Clock Divider High register

**Table 16-9. ICCLKH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved
15-0	ICCH15_ICCLH0	R/W	0h	High time I 2 C SCL Clock Division Factor. They are used to divide down the master clock to create the SCL high time transition frequency. This register must be configured while the I2C is still in reset (IRS_=0).

### 16.6.6 ICCNT Register (Offset = 14h) [Reset = 00000000h]

ICCNT is shown in [Table 16-10](#).

Return to the [Summary Table](#).

I2C Data Count register

**Table 16-10. ICCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved
15-0	ICDC15_ICDC0	R/W	0h	<p>Data count. This data count register is used to generate a Stop condition if a Stop condition is specified (STP=1).</p> <p>ICCNT=1 data count is 1 .....</p> <p>ICCNT=0FFFFh data count is 65535 ICCNT=0data counter is 65536 Note that ICCNT is a don't care when RM is set to 1.</p>

### 16.6.7 ICDRR Register (Offset = 18h) [Reset = 00000000h]

ICDRR is shown in [Table 16-11](#).

Return to the [Summary Table](#).

I2C Data Receive register

**Table 16-11. ICDRR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved
7-0	D7_D0	R/W	0h	Receive data

### 16.6.8 ICSAR Register (Offset = 1Ch) [Reset = 00000000h]

ICSAR is shown in [Table 16-12](#).

Return to the [Summary Table](#).

I2C Slave Address register

**Table 16-12. ICSAR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	NU	R/W	0h	Reserved
9-0	A9_A0	R/W	0h	Slave address. Use in both 7- and 10-bit address mode.

### 16.6.9 ICDXR Register (Offset = 20h) [Reset = 00000000h]

ICDXR is shown in [Table 16-13](#).

Return to the [Summary Table](#).

I2C Data Transmit register

**Table 16-13. ICDXR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved
7-0	D7_D0	R/W	0h	Transmit data



## 16.6.10 ICMR Register (Offset = 24h) [Reset = 00000000h]

ICMR is shown in [Table 16-14](#).

Return to the [Summary Table](#).

I2C Mode register

**Table 16-14. ICMR Register Field Descriptions**

Bit	Field	Type	Reset	Description																				
31-16	NU2	R/W	0h	Reserved																				
15	NACKMOD	R/W	0h	No Acknowledge (NACK) mode. This bit is used to send an Acknowledge (ACK) or a No Acknowledge (NACK) to the transmitter. This bit is only applicable when the I2C is in receiver mode. In master receiver mode when the internal data count counter decrements to zero the I2C sends a NACK. The master receiver I2C finishes a transfer when it sends a NACK. The I2C ignores ICCNT when NACKMOD is '1'. The NACKMOD bit should be set before the rising edge of the last data bit (bit 8) if a NACK must be sent and this bit is cleared once a NACK has been sent. NACKMOD=0 the I2C sends an ACK to the transmitter during the acknowledge cycle. NACKMOD=1 the I2C sends a NACK to the transmitter during the acknowledge cycle.																				
14	FREE	R/W	0h	Free Running. This bit is used to determine the state of the I2C when a breakpoint is encountered in the HLL debugger. FREE= 0: (default) Stops immediately if SCL is low and keep driving SCL low whether I2C is master transmitter/receiver. If SCL is high I2C waits until SCL becomes low and then stops. If the I2C is a slave it will stop when the transmission/receiving completes. FREE= 1: The I2C runs free.																				
13	STT	R/W	0h	Start Condition (Master only mode). This bit can be set to a"1" by the CPU to generate a Start condition. In master mode when setting Start to"1" generates a Start condition. It is reset to "0" by the hardware after the Start condition has been generated. The Start/Stop bits can be configured to generate different transfer formats. Note that the STT and STP can be used to terminate the repeat mode. <table><tr><th>STT</th><th>STP</th><th>Conditions</th><th>Bus Activities</th></tr><tr><td>1</td><td>0</td><td>Start</td><td>S-A-D</td></tr><tr><td>0</td><td>1</td><td>Stop</td><td>P</td></tr><tr><td>1</td><td>1</td><td>Start-Stop (ICCNT= n)</td><td>S-A-D..(n)..D-P</td></tr><tr><td>1</td><td>0</td><td>Start (ICCNT= n)</td><td>S-A-D..(n)..D</td></tr></table>	STT	STP	Conditions	Bus Activities	1	0	Start	S-A-D	0	1	Stop	P	1	1	Start-Stop (ICCNT= n)	S-A-D..(n)..D-P	1	0	Start (ICCNT= n)	S-A-D..(n)..D
STT	STP	Conditions	Bus Activities																					
1	0	Start	S-A-D																					
0	1	Stop	P																					
1	1	Start-Stop (ICCNT= n)	S-A-D..(n)..D-P																					
1	0	Start (ICCNT= n)	S-A-D..(n)..D																					
12	NU1	R/W	0h	Reserved for IDLEEN (IDLE Enable on 5509). - (RW )																				
11	STP	R/W	0h	Stop Condition (Master mode only). This bit can be set to a"1" by the CPU to generate a Stop condition. It is reset to "0" by the hardware after the Stop condition has been generated. The Stop condition is generated when ICCNT passes 0 when the I2C is in non-repeat mode(RM=0).																				

**Table 16-14. ICMR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
10	MST	R/W	0h	Master. MST= 0: The I2C peripheral is in the "slave" mode and clock is received from the "master" device. MST= 1: The I2C peripheral is in the "master" mode and it generates the clock. This bit is clear when the transfer completed.
9	TRX	R/W	0h	Transmitter. TRX= 0: The I2C is in the "receiver" mode and data on data line SDA is shifted into the data register ICDRR. TRX= 1: The I2C is in the "transmitter" mode and the data in ICDXR is shifted out on data line SDA. The operating modes (not in FDF mode) are defined as follows. In FDF mode TRX must be configured even if the I2C is in slave mode because there is no address/direction byte in FDF mode. <div> MST TRX Operating  Modes 0 x "slave receiver"  0 x "slave transmitter" 1 0 "master receiver" 1 1 "master transmitter" </div>
8	XA	R/W	0h	Expanded Address. XA= 0: (default) 7-bit address mode (normal address mode). XA= 1: 10-bit address mode (expanded address mode) Please note that XA needs to be configured even if the I2C is in slave mode.
7	RM	R/W	0h	Repeat Mode. This bit is set to a "1" by the CPU to put the I2C in the repeat mode. In this mode data is continuously transmitted out of the ICDXR until the STP bit is set to "1" regardless of ICCNT value. This bit is don't care if the I2C is configured in slave mode. <div> RM STT STP Conditions Bus  Activities Mode  0 0 0 Idle None NA  0 0 1 Stop P NA  0 1 0 (Re)Start S-A-D..(n)..D Repeat n  0 1 1 (Re)Start-Stop S-A-D..(n)..D-P Repeat  n 1 0 0 Idle none NA  1 0 1 Stop P NA  1 1 0 (Re)Start S-A-D-D-  D.. Continuous  1 1 1 Reserved None NA </div>
6	DLB	R/W	0h	Digital Loop Back (in master transmit mode only). This bit is set to a "1" by the CPU to put the I2C in the loop back mode. In this mode data transmitted out of the ICDXR will be received in the ICDRR after ((CPU freq/I2C freq)8) CPU cycles via an internal path. The address of the ICOAR is output on SDA.
5	IRS	R/W	0h	I2C Reset Not. This can be set to a "0" by the CPU to put the I2C in reset or to a "1" to take the I2C out of reset. When this bit is reset to 0 all status bits in ICSTR and ICIVR are set to default values. Note that if this bit is reset during a transfer it can cause the I2C bus hang (SDA and SCL are tri-stated).

**Table 16-14. ICMR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
4	STB	R/W	0h	Start Byte (Master only mode). The Start Byte mode bit is set to 1 by the CPU to configure the I2C in Start byte mode the I2C sends "00000001" regardless ICSAR value. Refer to the Philip I2C spec for more details.
3	FDF	R/W	0h	Free Data Format. This bit can be set to "1" by the CPU to configure the I2C in Free Data Format mode.  <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> FDF__MST__TRX__Operating mode__0__0__  x__Slave in non FDF mode__0__1__0__Master  receive in non FDF mode__0__1__1__Master  transmit in non FDF mode__1__0__0__Slave  receiver in FDF mode__1__0__1__Slave transmitter  in FDF mode__1__1__0__Master receiver in FDF  mode__1__1__1__Master transmitter in FDF mode </div>
2-0	BC2_BC1_BC0	R/W	0h	Bit Count : Bit Count 2, Bit Count 1 and Bit Count 0 define the number of bits starting from the lsb (excluding the acknowledge bit) of the next byte which are yet to be received or transmitted.  <div style="border: 1px solid black; padding: 5px; margin: 5px 0;"> BC2_BC1_BC0__Bits/byte in FDF__Bits/byte w/  ACK__0__0__1__NA (reserved)__NA  (reserved)__0__1__0__2__3__  0__1__1__3__4__  1__0__0__4__5__  1__0__1__5__6__  1__1__0__6__7__  1__1__1__7__8__  0__0__0__8__9__ </div>

### 16.6.11 ICIVR Register (Offset = 28h) [Reset = 00000000h]

ICIVR is shown in [Table 16-15](#).

Return to the [Summary Table](#).

I2C Interrupt Vector register

**Table 16-15. ICIVR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	NU2	R/W	0h	Reserved.
11-8	TESTMD	R/W	0h	Reserved for internal testing.
7-3	NU1	R/W	0h	Reserved.
2-0	INTCODE	R/W	0h	<p>Interrupt code. The binary-coded-interrupt vector indicates which interrupt has occurred. Reading the ICIVR clears the interrupt code except ARDY(011) RRDY(100) and XRDY(101). Interrupt code for ARDY RRDY and XRDY is cleared when ARDY ICRRDY and ICXRDY bits in the ICSTR is cleared to default value respectively. If other interrupts are pending a new interrupt is generated. If there are more than one interrupt flag reading the ICIVR clears the highest priority interrupt code. Reading the ICIVR also clears corresponding status bit in the ICSTR except ARDY ICRRDY ICXRDY and AAS. Note that users must read (clear) the ICIVR before doing another start otherwise the ICIVR could contain incorrect (old interrupt flags) value.</p> <p> Interrupt Code _____ Interrupt Occurred _____  _000_ (default) _____ None _001_ (highest  priority) _____ Arbitration Lost interrupt _010 _____ No  Acknowledgement interrupt _011 _____ Register  Access Ready interrupt _100 _____ Receive Data  Ready interrupt _101 _____ Transmit Data Ready  interrupt _110 _____ Stop Condition Detection  _111_ (lowest priority) _____ Address As Slave - (RW) </p>

### 16.6.12 ICEMDR Register (Offset = 2Ch) [Reset = 00000000h]

ICEMDR is shown in [Table 16-16](#).

Return to the [Summary Table](#).

I2C Extended Mode register

**Table 16-16. ICEMDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved. - (RW )
1	IGNACK	R/W	0h	Ignore NACK mode IGNACK=0 The master transmitter will operate normally discontinue the data transfer and set the ARDY and NACK status bits when a NACK signal is received from the slave. IGNACK=1 The master transmitter will ignore a NACK received from the slave.
0	BCM	R/W	0h	Backward Compatibility Mode. This bit affects the I2C interrupt behavior. Refer to appendix A for details.

### 16.6.13 ICPSC Register (Offset = 30h) [Reset = 00000000h]

ICPSC is shown in [Table 16-17](#).

Return to the [Summary Table](#).

I2C Prescaler register

**Table 16-17. ICPSC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved.
7-0	IPSC7_IPSC0	R/W	0h	8-bit prescaler to divide the system clock down to 4/8/12Mhz clock and used by the I2C module. This register must be initialized while the I2C is still in reset (IRS_ =0). The value takes effect on the rising edge of IRS_.

### 16.6.14 ICPID1 Register (Offset = 34h) [Reset = 00000000h]

ICPID1 is shown in [Table 16-18](#).

Return to the [Summary Table](#).

I2C Peripheral ID register 1

**Table 16-18. ICPID1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU	R/W	0h	Reserved.
15-8	CLASS	R/W	0h	Identifies the class of peripheral. This value should be 0x01 - (RW )
7-0	REVISION	R/W	0h	Identifies the revision level of the I2C. This value should be incremented each time the design is revised. - (RW )

### 16.6.15 ICPID2 Register (Offset = 38h) [Reset = 00000000h]

ICPID2 is shown in [Table 16-19](#).

Return to the [Summary Table](#).

I2C Peripheral ID register 2

**Table 16-19. ICPID2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved.
7-0	TYPE	R/W	0h	Identifies the type of peripheral. This value should be 0x05 - (RW )



## 16.6.16 ICDMAC Register (Offset = 3Ch) [Reset = 00000000h]

ICDMAC is shown in [Table 16-20](#).

Return to the [Summary Table](#).

I2C DMA Control Register

**Table 16-20. ICDMAC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved. - (RW )
1	TXDMAEN	R/W	0h	Transmit DMA enable. This bit controls the receive DMA event pin to the system. When this bit is 1 the DMA event is enabled and ICTEVT_POR pin is asserted when the DMA transfer is required. When this bit is 0 the ICTEVT_POR pin is never asserted. RXDMAEN= 0: DMA transmit event is disabled. RXDMAEN= 1: DMA transmit event is enabled. (Default)
0	RXDMAEN	R/W	0h	Receive DMA enable. This bit controls the receive DMA event pin to the system. When this bit is 1 the DMA event is enabled and ICREVT_POR pin is asserted when the DMA transfer is required. When this bit is 0 the ICREVT_POR pin is never asserted. RXDMAEN= 0: DMA receive event is disabled. RXDMAEN= 1: DMA receive event is enabled. (Default)

### 16.6.17 I2C\_RESERVED1 Register (Offset = 40h) [Reset = 00000000h]

I2C\_RESERVED1 is shown in [Table 16-21](#).

Return to the [Summary Table](#).

Reserved

**Table 16-21. I2C\_RESERVED1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Reserved.

### 16.6.18 I2C\_RESERVED2 Register (Offset = 44h) [Reset = 00000000h]

I2C\_RESERVED2 is shown in [Table 16-22](#).

Return to the [Summary Table](#).

Reserved

**Table 16-22. I2C\_RESERVED2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU	R/W	0h	Reserved.

### 16.6.19 ICPFUNC Register (Offset = 48h) [Reset = 00000000h]

ICPFUNC is shown in [Table 16-23](#).

Return to the [Summary Table](#).

I2C Pin Function register

**Table 16-23. ICPFUNC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	NU	R/W	0h	Reserved.
0	PFUNC0	R/W	0h	Controls the function of the I2C SCL and SDA pins. 0 = Pins function as SCL and SDA 1 = Pins functions as GPIO Note: No hardware protection is required to disable I2C function when the PFUNC[0] and IRS_ bits are both set to one. When PFUNC[0] is "1" (GPIO mode) the sub-module which controls the I2C function receives the value "1" for SCL and SDA. IRS_ can be set to "1" regardless of PFUNC[0] and the I2C function works whenever the IRS_ bit is "1". The user is expected to hold I2C in reset via IRS_ bit when changing to/from GPIO mode via the PFUNC[0] bit.

### 16.6.20 ICPDIR Register (Offset = 4Ch) [Reset = 00000000h]

ICPDIR is shown in [Table 16-24](#).

Return to the [Summary Table](#).

I2C Pin Direction register

**Table 16-24. ICPDIR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDIR1	R/W	0h	Controls the direction of the I2C SDA pin when configured as GPIO. 0 = SDA pin functions as input 1 = SDA pin functions as output
0	PDIR0	R/W	0h	Controls the direction of the I2C SCL pin when configured as GPIO. 0 = SCL pin functions as input 1 = SCL pin functions as output

### 16.6.21 ICPDIN Register (Offset = 50h) [Reset = 00000000h]

ICPDIN is shown in [Table 16-25](#).

Return to the [Summary Table](#).

I2C Pin Data In register

**Table 16-25. ICPDIN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDIN1	R/W	0h	Indicates the logic level present on the SDA pin. Reads: 0 = Logic low present at SDA pin regardless of PFUNC setting. 1 = Logic high present at SDA pin regardless of PFUNC setting. Writes: Writes have no effect. - (RW )
0	PDIN0	R/W	0h	Indicates the logic level present on the SCL pin. Reads: 0 = Logic low present at SCL pin regardless of PFUNC setting. 1 = Logic high present at SCL pin regardless of PFUNC setting. Writes: Writes have no effect - (RW )

## 16.6.22 ICPDOUT Register (Offset = 54h) [Reset = 00000000h]

ICPDOUT is shown in [Table 16-26](#).

Return to the [Summary Table](#).

I2C Pin Data Out register

**Table 16-26. ICPDOUT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDOUT1	R/W	0h	Controls the level driven on the SDA pin when configured as GPIO output. Reads: Reads return register values not GPIO pin levels. Writes: 0 = SDA pin driven low 1 = SDA pin driven high. Note: If SDA is connected to an open-drain buffer at the chip level the I2C cannot drive SDA to high.
0	PDOUT0	R/W	0h	Controls the level driven on the SCL pin when configured as GPIO output. Reads: Reads return register values not GPIO pin levels. Writes: 0 = SCL pin driven low 1 = SCL pin driven high Note: If SCL is connected to an open-drain buffer at the chip level the I2C cannot drive SCL to high.

### 16.6.23 ICPDSET Register (Offset = 58h) [Reset = 00000000h]

ICPDSET is shown in [Table 16-27](#).

Return to the [Summary Table](#).

I2C Pin Data Set register

**Table 16-27. ICPDSET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDSET1	R/W	0h	Used to set PDOUT[1] bit which corresponds to the SDA GPIO pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[1] bit is set to logic high.
0	PDSET0	R/W	0h	Used to set PDOUT[0] bit which corresponds to the SCL GPIO pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[0] bit is set to logic high.



## 16.6.24 ICPDCLR Register (Offset = 5Ch) [Reset = 00000000h]

ICPDCLR is shown in [Table 16-28](#).

Return to the [Summary Table](#).

I2C Pin Data Clear register

**Table 16-28. ICPDCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDCLR1	R/W	0h	Used to clear PDOUT[1] bit which corresponds to the SDA pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[1] bit is cleared to logic low.
0	PDCLR0	R/W	0h	Used to clear PDOUT[0] bit which corresponds to the SCL pin. Reads: Reads should return 0. User documentation should say reads are indeterminate. Writes: 0 = no effect 1 = PDOUT[0] bit is cleared to logic low.

### 16.6.25 ICPDRV Register (Offset = 60h) [Reset = 00000000h]

ICPDRV is shown in [Table 16-29](#).

Return to the [Summary Table](#).

I2C Pin Driver Mode Register

**Table 16-29. ICPDRV Register Field Descriptions**

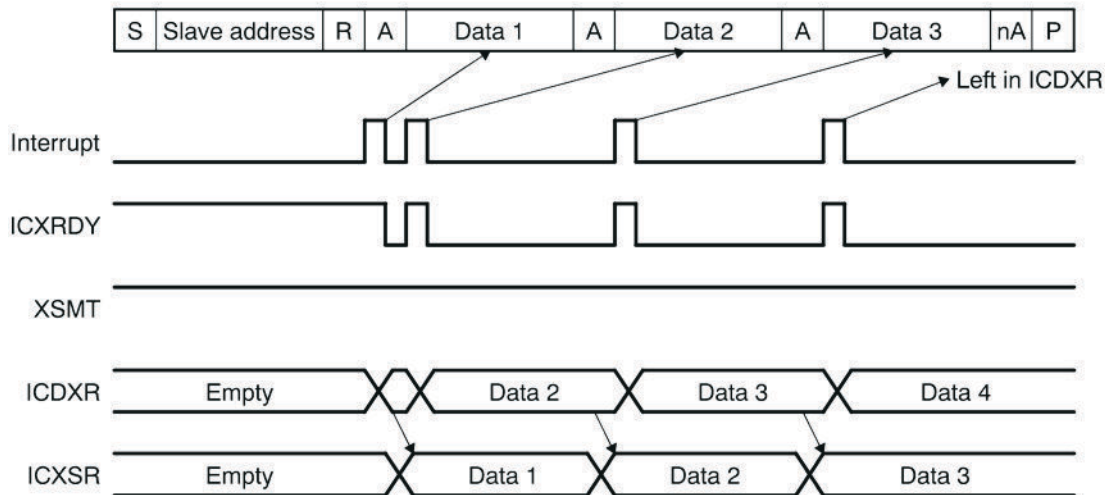
Bit	Field	Type	Reset	Description
31-2	NU	R/W	0h	Reserved
1	PDRV1	R/W	0h	Used to select driver mode of output buffer for SDA pin. 0 = I2C mode. 1 = GPIO mode. Note: Value of this register is reflected on the PDRV_SDA_POR port. Actual function depends on I/O buffer and chip implementation.
0	PDRV0	R/W	0h	Used to select driver mode of output buffer for SCL pin. 0 = I2C mode. 1 = GPIO mode. Note: Value of this register is reflected on the PDRV_SCL_POR port. Actual function depends on I/O buffer and chip implementation.

## 16.7 Sample Waveforms

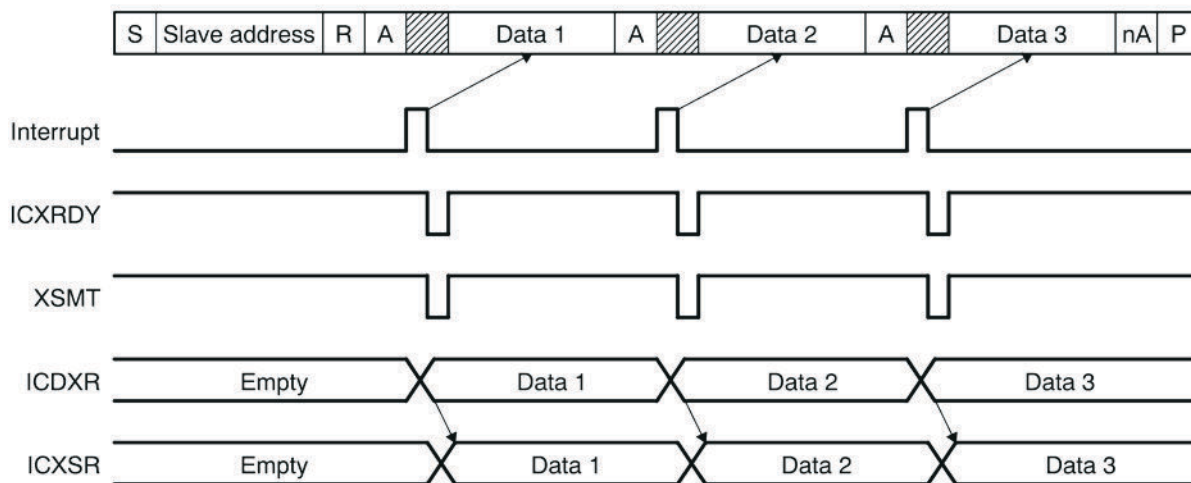
Figure 16-13 provides waveforms to illustrate the difference between normal operation and backward compatibility mode.

Slave transmitter

a) BCM=1



b) BCM=0



**Figure 16-13. Difference between Normal Operation and Backward Compatibility Mode**

## Chapter 17

# General-Purpose Input/Output (GPIO) Module

---



This chapter describes the general-purpose input/output (GPIO) module. The GPIO module provides the family of devices with input/output (I/O) capability. The I/O pins are bidirectional and bit-programmable. The GPIO module also supports external interrupt capability.

---

### Note

The "GPIO" module is also known as the "GPIO" module in other TI MCU and MPU devices. The two terms are used interchangeably and represent the general use I/O module of the device.

---

## 17.1 Overview

The GPIO module offers general-purpose input and output capability. It supports up to eight 8-bit ports for a total of up to 64 GPIO terminals. Each of these 64 terminals can be independently configured as input or output and configured as required by the application. The GPIO module also supports generation of interrupts whenever a rising edge or falling edge or any toggle is detected on up to 32 of these GPIO terminals. Refer to the device datasheet for identifying the number of GPIO ports supported and the GPIO terminals capable of generating an interrupt.

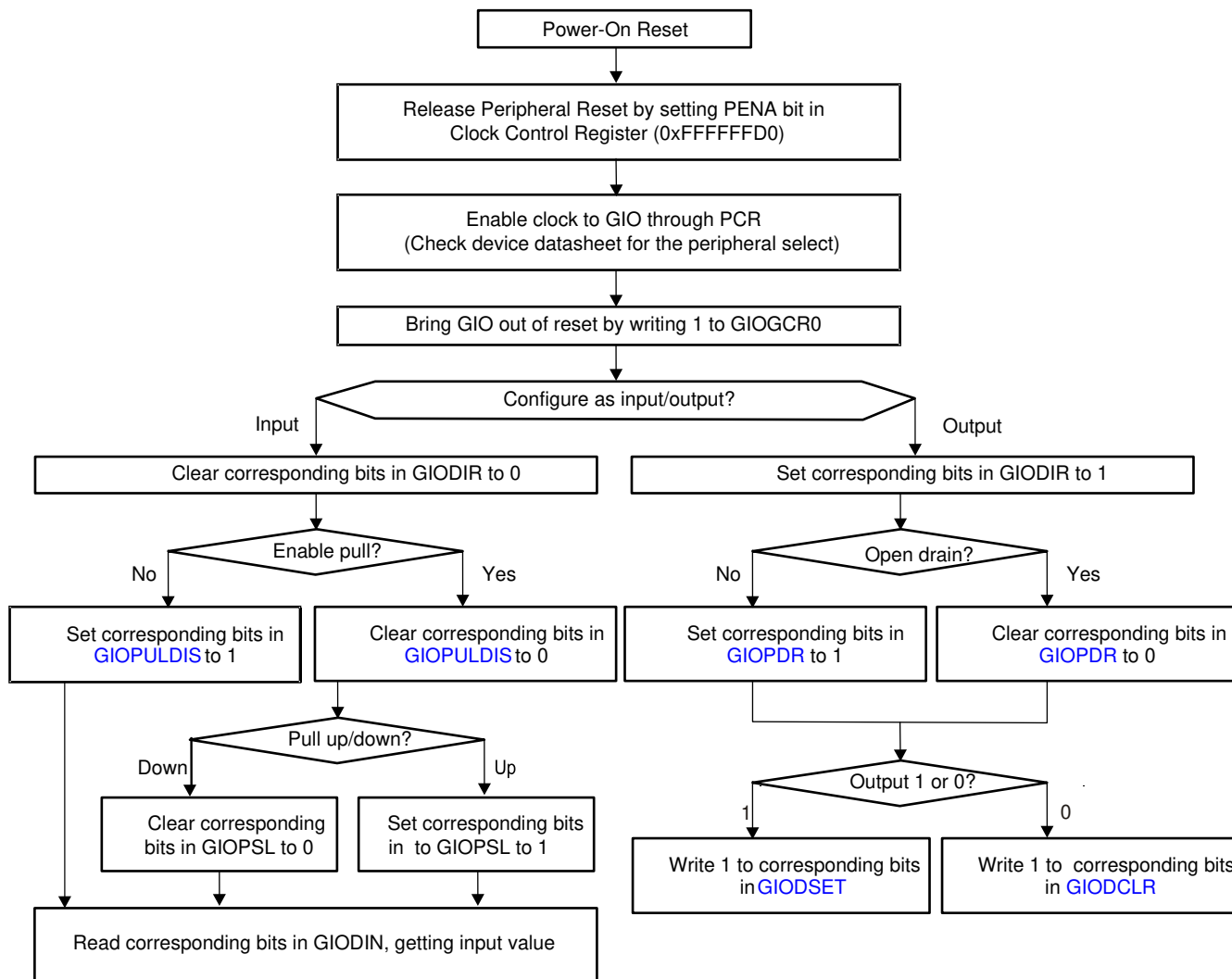
The main features of the GPIO module are summarized as follows:

- Allows each GPIO terminal to be configured for general-purpose input or output functions
- Supports programmable pull directions on each input GPIO terminal
- Supports GPIO output in push/pull or open-drain modes
- Allows up to 32 GPIO terminals to be used for generating interrupt requests

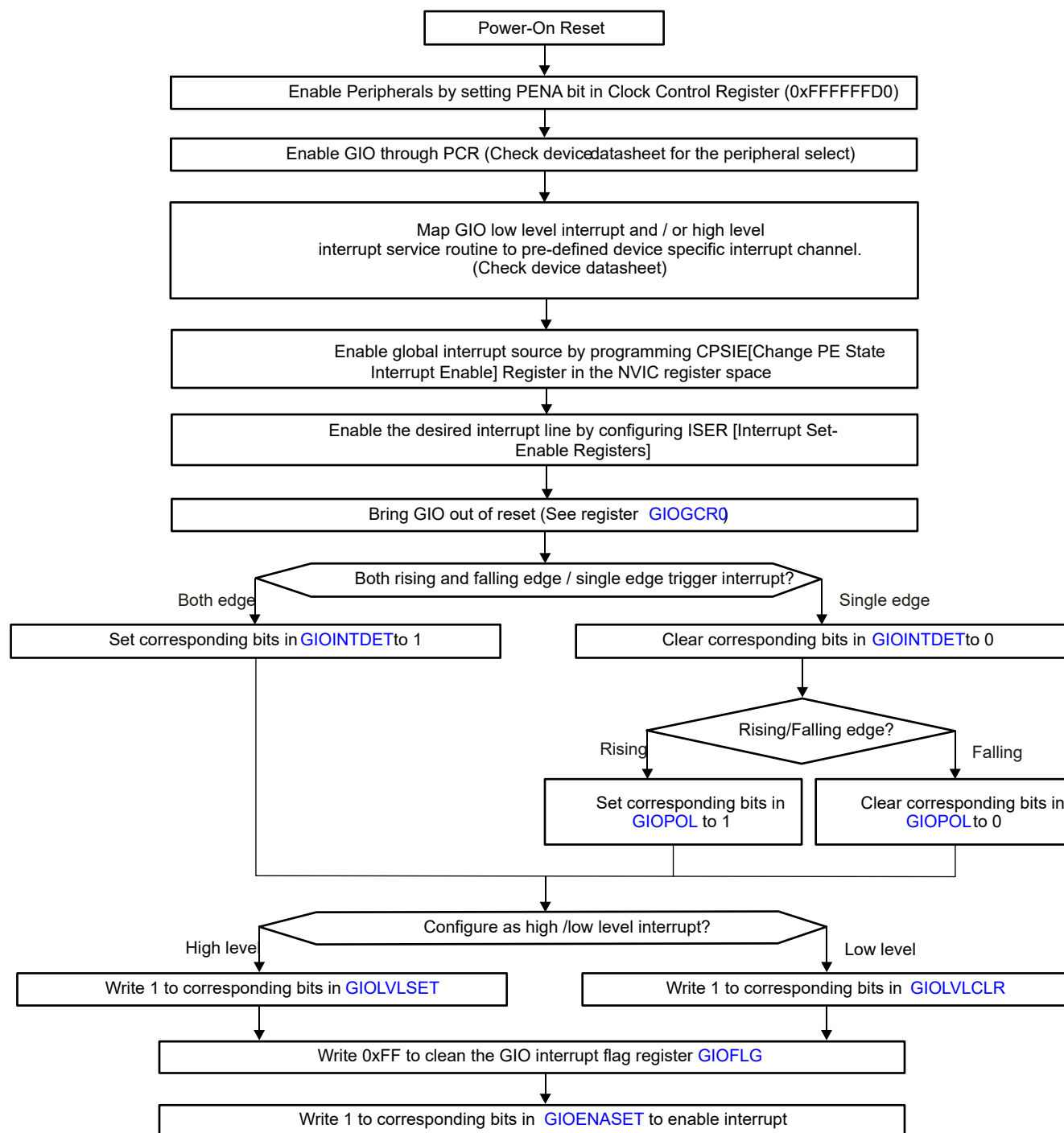
## 17.2 Quick Start Guide

The GPIO module comprises two separate components: an input/output (I/O) block and an interrupt generation block. [Figure 17-1](#) and [Figure 17-2](#) show what you should do after reset to configure the GPIO module as I/O or for generating interrupts.

In GPIO interrupt service routine, you shall read the GPIO offset register (GIOFF1 or GIOFF2, depending on high-/low-level interrupt) to clear the flag and find the pending interrupt GPIO channel.

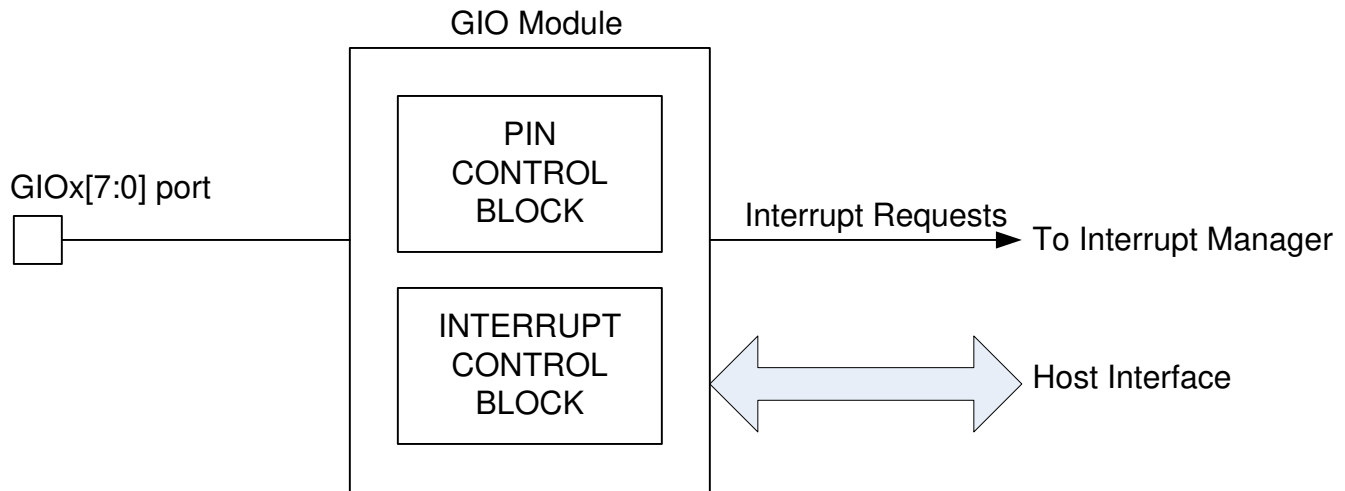


**Figure 17-1. I/O Function Quick Start Flow Chart**


**Figure 17-2. Interrupt Generation Function Quick Start Flow Chart**

### 17.3 Functional Description of GPIO Module

As shown in Figure 17-3, the GPIO module comprises of two separate components: an input/output (I/O) block and an interrupt block.



**Figure 17-3. GPIO Module Diagram**

#### 17.3.1 I/O Functions

The I/O block allows each GPIO terminal to be configured for use as a general-purpose input or output in the application. The GPIO module supports multiple registers to control the various aspects of the input and output functions. These are described as follows.

- Data direction (GPIODIR)

Configures GPIO terminal(s) as input (default) or output through the GPIODIRx registers.

- Data input (GPIODIN)

Reflects the logic level on GPIO terminals in the GPIODINx registers. A high voltage ( $V_{IH}$  or greater) applied to the pin causes a high value (1) in the data input register (GPIODIN[7:0]). When a low voltage ( $V_{IL}$  or less) is applied to the pin, the data input register reads a low value (0). The  $V_{IH}$  and  $V_{IL}$  values are device specific and can be found in the device datasheet.

- Data output (GPIODOUT)

Configures the logic level to be output on GPIO terminal(s) configured as outputs. A low value (0) written to the data output register forces the pin to a low output voltage ( $V_{OL}$  or lower). A high value (1) written to the data output register (GPIODOUTx) forces the pin to a high output voltage ( $V_{OH}$  or higher) if the open drain functionality is disabled (GPIOPDRx[7:0]). If open drain functionality is enabled, a high value (1) written to the data output register forces the pin to a high-impedance state (Z).

- Data set (GPIODSET)

Allows logic HIGH to be output on GPIO terminal(s) configured as outputs by writing 1's to the required bits in the GPIODSETx registers. If open drain functionality is enabled, a high value (1) written to the data output register forces the pin to a high-impedance state (Z). The GPIODSETx registers eliminate the need for the application to perform a read-modify-write operation when it needs to set one or more GPIO pin(s).

- Data clear (GPIODCLR)

Allows logic LOW to be output on GPIO terminal(s) configured as outputs by writing 1s to the required bits in the GPIODCLR registers. The GPIODCLR registers eliminate the need for the application to perform a read-modify-write operation when it needs to clear one or more GPIO pin(s).

- Open drain (GPIOPDR)

Open drain functionality is enabled or disabled (default) using the open drain register `GIOPDR[7:0]` register. If open-drain mode output is enabled on a pin, a high value (1) written to the data output register (`GIODOUTx[7:0]`) forces the pin to a high impedance state (Z).

- Pull disable (`GIOPULDIS`)

Disables the internal pull on GPIO terminal(s) configured as inputs by writing to the `GIOPULDISx` registers.

- Pull select (`GIOPSL`)

Selects internal pull down (default) or pull up on GPIO terminal(s) configured as inputs by writing to the `GIOPULSELx` registers.

Refer to the specific device's datasheet to identify the number of GPIO ports as well as the input and output functions supported. Some devices may not support the programmable pull controls. In that case, the pull disable and the pull select register controls will not work.

### 17.3.2 Interrupt Function

The GPIO module supports up to 32 terminals to be configured for generating an interrupt to the host processor through the Interrupt Manager (NVIC/IM). The main functions of the interrupt block are:

- Select the GPIO pin(s) that is/are used to generate interrupt(s)

This is done via the interrupt enable set and clear registers, `GIOENASET` and `GIOENACLR`.

- Select the edge on the selected GPIO pin(s) that is/are used to generate interrupt(s): rising/falling/both

Rising or falling edge can be selected via the `GIOPOL` register. If interrupt is required to be generated on both rising and falling edges, this can be configured via the `GIOINTDET` register.

- Select the interrupt priority

Low- or high-level interrupt can be selected through the `GIOLVLSET` and `GIOLVLCLR` registers.

- Individual interrupt flags are set in the `GIOFLG` register

The terminals on GPIO ports A through D are all interrupt-capable and can be used to handle either general I/O functions or interrupt requests. Each interrupt request can be connected to the NVIC/IM at one of two different levels – High (or A) and Low (or B), depending on the NVIC/IM channel number. The NVIC/IM has an inherent priority scheme so that a request on a lower number channel has a higher priority than a request on a higher number channel. Refer the device datasheet to identify the NVIC/IM channel numbers for the GPIO level A and level B interrupt requests. Also note that the interrupt priority of level A and level B interrupt handling blocks can be re-programmed in the NVIC/IM.



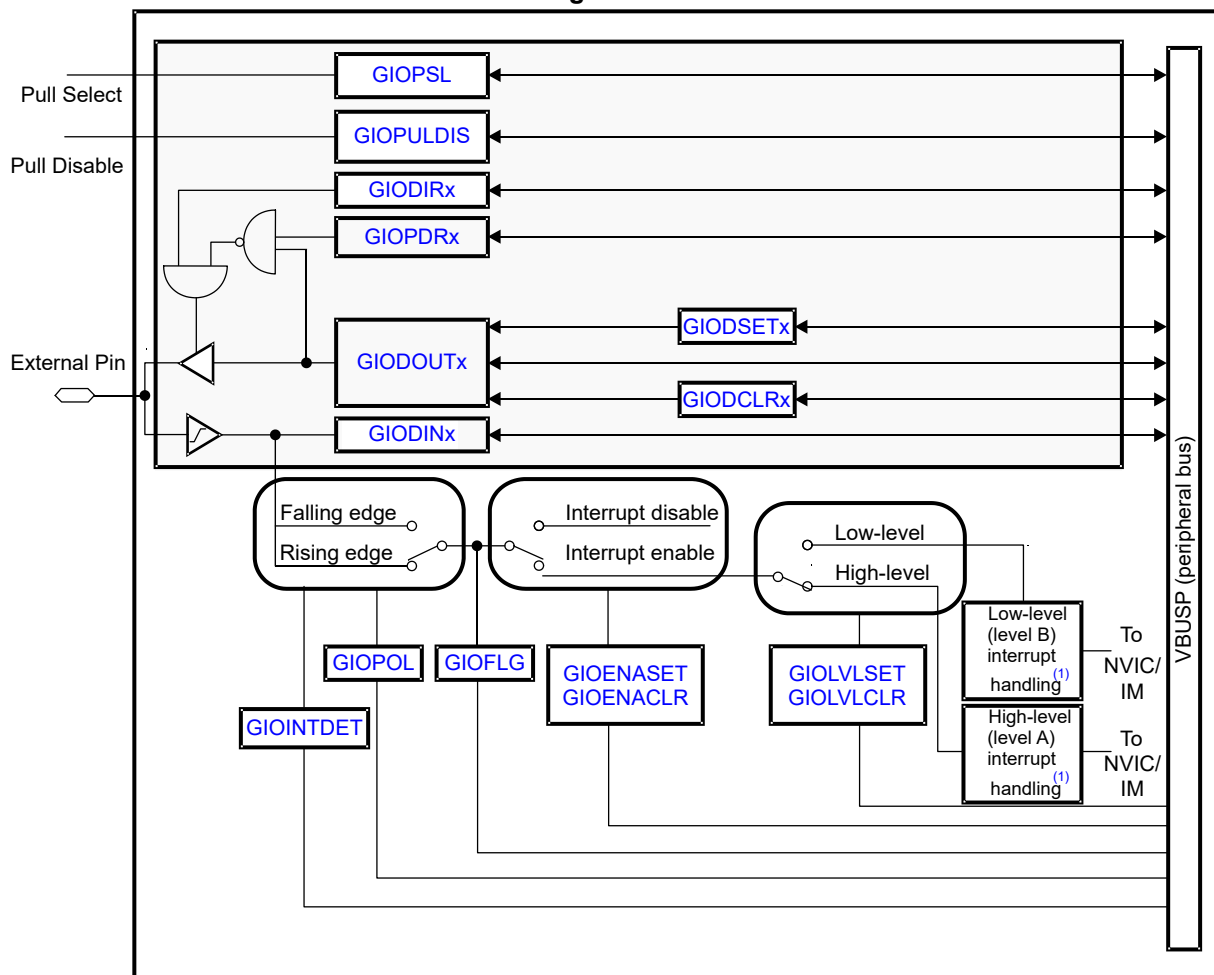
### 17.3.3 GPIO Block Diagram

The GPIO block diagram (Figure 17-4) represents the flow of information through a pin. The shaded area corresponds to the I/O block; the unshaded area corresponds to the interrupt block.

**Figure 17-4. GPIO Block Diagram**

- A. A single low-level-interrupt-handling block and a single high-level-interrupt-handling block service all of the interrupt-capable external pins, but only one pin can be serviced by an interrupt block at a time.

**Figure 17-5.**



## 17.4 Device Modes of Operation

The GPIO module behaves differently in different modes of operation. There are two main modes:

- Emulation mode
- Power-down mode (low-power mode)

### 17.4.1 Emulation Mode

Emulation mode is used by debugger tools to stop the CPU at breakpoints to read registers.

---

#### Note

##### Emulation Mode and Emulation Registers

Emulation mode is a mode of operation of the device and is separate from the GPIO emulation registers (GIOEMU1 and GIOEMU2). The contents of these emulation registers are identical to the contents of GPIO offset registers (GIOOFF1 and GIOOFF2). Both emulation registers and GPIO offset registers are NOT cleared when they are read in emulation mode. GPIO offset registers are cleared when they are read in normal mode (other than emulation mode). The emulation registers are NOT cleared when they are read in normal mode. The intention for the emulation registers is that software can use them without clearing the flags.

---

During emulation mode:

- External interrupts are not captured because the NVIC/IM is unable to service interrupts.
- Any register can be read without affecting the state of the system.
- A write to a register still does affect the state of the system.

### 17.4.2 Power-Down Mode (Low-Power Mode)

In power-down mode, the clock signal to the GPIO module is disabled. Thus, there is no switching and the only current draw comes from leakage current. In power-down mode, interrupt pins become level-sensitive rather than edge-sensitive. The polarity bit changes function from falling-edge-triggered to low-level-triggered and rising-edge-triggered to high-level-triggered. A corresponding level on an interrupt pin pulls the module out of low-power mode, if the interrupt is also enabled to wake up the device out of a low-power mode.

### 17.4.3 Interrupts

GPIO generates aggregated interrupts for all inputs from PAD. Some of the interrupts to MSS\_R5F are directly taken from GPIO-PAD.

MSS\_GPIO\_INT0/1

MSS\_GPIO\_PAD\_INT0 from GPIO[0]

MSS\_GPIO\_PAD\_INT1 from GPIO[1]

MSS\_GPIO\_PAD\_INT2 from GPIO[2]

MSS\_CTRL:: MSS\_GPIO\_CFG\_GPIO:: MSS\_GPIO\_CFG\_GPIO\_CONFIG is used for selecting the edge that can give a trigger.

## 17.5 TOP\_GPIO Registers

Table 17-1 lists the memory-mapped registers for the TOP\_GPIO registers. All register offset addresses not listed in Table 17-1 should be considered as reserved locations and the register contents should not be modified.

**Table 17-1. TOP\_GPIO Registers**

Offset	Acronym	Register Name	Section
0h	GIOGCR	GIOGCR	<a href="#">Go</a>
4h	GIOPWDN	GIOPWDN	<a href="#">Go</a>
8h	GIOINTDET	GIOINTDET	<a href="#">Go</a>
Ch	GIOPOL	GIOPOL	<a href="#">Go</a>
10h	GIOENASET	GIOENASET	<a href="#">Go</a>
14h	GIOENACLR	GIOENACLR	<a href="#">Go</a>
18h	GIOLVLSET	GIOLVLSET	<a href="#">Go</a>
1Ch	GIOLVLCLR	GIOLVLCLR	<a href="#">Go</a>
20h	GIOFLG	GIOFLG	<a href="#">Go</a>
24h	GIOOFFA	GIOOFFA	<a href="#">Go</a>
28h	GIOOFFB	GIOOFFB	<a href="#">Go</a>
2Ch	GIOEMUA	GIOEMUA	<a href="#">Go</a>
30h	GIOEMUB	GIOEMUB	<a href="#">Go</a>
34h	GIODIRA	GIODIRA	<a href="#">Go</a>
38h	GIODINA	GIODINA	<a href="#">Go</a>
3Ch	GIODOUTA	GIODOUTA	<a href="#">Go</a>
40h	GIOSETA	GIOSETA	<a href="#">Go</a>
44h	GIOCLRA	GIOCLRA	<a href="#">Go</a>
48h	GIOPDRA	GIOPDRA	<a href="#">Go</a>
4Ch	GIOPULDISA	GIOPULDISA	<a href="#">Go</a>
50h	GIOPSLA	GIOPSLA	<a href="#">Go</a>
54h	GIODIRB	GIODIRB	<a href="#">Go</a>
58h	GIODINB	GIODINB	<a href="#">Go</a>
5Ch	GIODOUTB	GIODOUTB	<a href="#">Go</a>
60h	GIOSETB	GIOSETB	<a href="#">Go</a>
64h	GIOCLRB	GIOCLRB	<a href="#">Go</a>
68h	GIOPDRB	GIOPDRB	<a href="#">Go</a>
6Ch	GIOPULDISB	GIOPULDISB	<a href="#">Go</a>
70h	GIOPSLB	GIOPSLB	<a href="#">Go</a>
74h	GIODIRC	GIODIRC	<a href="#">Go</a>
78h	GIODINC	GIODINC	<a href="#">Go</a>
7Ch	GIODOUTC	GIODOUTC	<a href="#">Go</a>
80h	GIOSETC	GIOSETC	<a href="#">Go</a>
84h	GIOCLRC	GIOCLRC	<a href="#">Go</a>
88h	GIOPDRC	GIOPDRC	<a href="#">Go</a>
8Ch	GIOPULDISC	GIOPULDISC	<a href="#">Go</a>
90h	GIOPSLC	GIOPSLC	<a href="#">Go</a>
94h	GIODIRD	GIODIRD	<a href="#">Go</a>
98h	GIODIND	GIODIND	<a href="#">Go</a>
9Ch	GIODOUTD	GIODOUTD	<a href="#">Go</a>
A0h	GIOSETD	GIOSETD	<a href="#">Go</a>

**Table 17-1. TOP\_GIO Registers (continued)**

Offset	Acronym	Register Name	Section
A4h	GIOCLRD	GIOCLRD	<a href="#">Go</a>
A8h	GIOPDRD	GIOPDRD	<a href="#">Go</a>
ACh	GIOPULDISD	GIOPULDISD	<a href="#">Go</a>
B0h	GIOPSLD	GIOPSLD	<a href="#">Go</a>
B4h	GIODIRE	GIODIRE	<a href="#">Go</a>
B8h	GIODINE	GIODINE	<a href="#">Go</a>
BCh	GIODOUTE	GIODOUTE	<a href="#">Go</a>
C0h	GIOSETE	GIOSETE	<a href="#">Go</a>
C4h	GIOCLRE	GIOCLRE	<a href="#">Go</a>
C8h	GIOPDRE	GIOPDRE	<a href="#">Go</a>
CCh	GIOPULDISE	GIOPULDISE	<a href="#">Go</a>
D0h	GIOPSLE	GIOPSLE	<a href="#">Go</a>
D4h	GIODIRF	GIODIRF	<a href="#">Go</a>
D8h	GIODINF	GIODINF	<a href="#">Go</a>
DCh	GIODOUTF	GIODOUTF	<a href="#">Go</a>
E0h	GIOSETF	GIOSETF	<a href="#">Go</a>
E4h	GIOCLRf	GIOCLRf	<a href="#">Go</a>
E8h	GIOPDRF	GIOPDRF	<a href="#">Go</a>
ECh	GIOPULDISF	GIOPULDISF	<a href="#">Go</a>
F0h	GIOPSLF	GIOPSLF	<a href="#">Go</a>
F4h	GIODIRG	GIODIRG	<a href="#">Go</a>
F8h	GIODING	GIODING	<a href="#">Go</a>
FCh	GIODOUTG	GIODOUTG	<a href="#">Go</a>
100h	GIOSETG	GIOSETG	<a href="#">Go</a>
104h	GIOCLRG	GIOCLRG	<a href="#">Go</a>
108h	GIOPDRG	GIOPDRG	<a href="#">Go</a>
10Ch	GIOPULDISG	GIOPULDISG	<a href="#">Go</a>
110h	GIOPSLG	GIOPSLG	<a href="#">Go</a>
114h	GIODIRH	GIODIRH	<a href="#">Go</a>
118h	GIODINH	GIODINH	<a href="#">Go</a>
11Ch	GIODOUTH	GIODOUTH	<a href="#">Go</a>
120h	GIOSETH	GIOSETH	<a href="#">Go</a>
124h	GIOCLRh	GIOCLRh	<a href="#">Go</a>
128h	GIOPDRH	GIOPDRH	<a href="#">Go</a>
12Ch	GIOPULDISH	GIOPULDISH	<a href="#">Go</a>
130h	GIOPSLH	GIOPSLH	<a href="#">Go</a>
134h	GIOSRCA	GIOSRCA	<a href="#">Go</a>
138h	GIOSRCB	GIOSRCB	<a href="#">Go</a>
13Ch	GIOSRCC	GIOSRCC	<a href="#">Go</a>
140h	GIOSRCD	GIOSRCD	<a href="#">Go</a>
144h	GIOSRCE	GIOSRCE	<a href="#">Go</a>
148h	GIOSRCF	GIOSRCF	<a href="#">Go</a>
14Ch	GIOSRCG	GIOSRCG	<a href="#">Go</a>
150h	GIOSRCH	GIOSRCH	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 17-2](#) shows the codes that are used for access types in this section.

**Table 17-2. TOP\_GPIO Access Type Codes**

Access Type	Code	Description
<b>Read Type</b>		
R	R	Read
<b>Write Type</b>		
W	W	Write
<b>Reset or Default Value</b>		
-n		Value after reset or the default value

### 17.5.1 GLOGCR Register (Offset = 0h) [Reset = 00000000h]

GLOGCR is shown in [Table 17-3](#).

Return to the [Summary Table](#).

GPIO reset

**Table 17-3. GLOGCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	NU0	R/W	0h	Reserved
0	RESET	R/W	0h	GPIO reset

### 17.5.2 GIOPWDN Register (Offset = 4h) [Reset = 00000000h]

GIOPWDN is shown in [Table 17-4](#).

Return to the [Summary Table](#).

GPIO power down mode register

**Table 17-4. GIOPWDN Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	NU	R/W	0h	Reserved
0	GIOPWDN	R/W	0h	<p>Writing to the GIOPWDN bit is only allowed in privilege mode. Reading of the GIOPWDN bit is allowed in all modes.</p> <p>Privilege mode (write):</p> <p>0 = Normal operation clocks enabled to GPIO module</p> <p>1 = Power-down mode User mode (write): Writes have no effect in user mode.</p> <p>User or privilege mode (read):</p> <p>0 = Normal operation clocks enabled to GPIO module</p> <p>1 = Power-down mode</p>

### 17.5.3 GIOINTDET Register (Offset = 8h) [Reset = 00000000h]

GIOINTDET is shown in [Table 17-5](#).

Return to the [Summary Table](#).

Interrupt detection select for pins [0:1] GIO[7:0].

**Table 17-5. GIOINTDET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIOINTDET_3	R/W	0h	Interrupt detection select for pins GIOD [7:0].
23-16	GIOINTDET_2	R/W	0h	Interrupt detection select for pins GIOC [7:0].
15-8	GIOINTDET_1	R/W	0h	Interrupt detection select for pins GIOB [7:0].
7-0	GIOINTDET_0	R/W	0h	Interrupt detection select for pins GIOA [7:0].



### 17.5.4 GIOPOL Register (Offset = Ch) [Reset = 00000000h]

GIOPOL is shown in [Table 17-6](#).

Return to the [Summary Table](#).

Interrupt polarity select for pins [0:1] GIO[7:0].

**Table 17-6. GIOPOL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIOPOL_3	R/W	0h	Interrupt polarity select for pins GIOD [7:0]
23-16	GIOPOL_2	R/W	0h	Interrupt polarity select for pins GIOC [7:0]
15-8	GIOPOL_1	R/W	0h	Interrupt polarity select for pins GIOB [7:0]
7-0	GIOPOL_0	R/W	0h	Interrupt polarity select for pins GIOA [7:0]

### 17.5.5 GIOENASET Register (Offset = 10h) [Reset = 00000000h]

GIOENASET is shown in [Table 17-7](#).

Return to the [Summary Table](#).

Interrupt enable for pins [0:1] GIO[7:0].

**Table 17-7. GIOENASET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIOENASET_3	R/W	0h	Interrupt enable for pins GIOD [7:0]
23-16	GIOENASET_2	R/W	0h	Interrupt enable for pins GIOC [7:0]
15-8	GIOENASET_1	R/W	0h	Interrupt enable for pins GIOB [7:0]
7-0	GIOENASET_0	R/W	0h	Interrupt enable for pins GIOA [7:0]

### 17.5.6 GIOENACLR Register (Offset = 14h) [Reset = 00000000h]

GIOENACLR is shown in [Table 17-8](#).

Return to the [Summary Table](#).

Interrupt enable for pins [0:1] GIO[7:0].

**Table 17-8. GIOENACLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIOENACLR_3	R/W	0h	Interrupt enable for pins GIOD [7:0]
23-16	GIOENACLR_2	R/W	0h	Interrupt enable for pins GIOC [7:0]
15-8	GIOENACLR_1	R/W	0h	Interrupt enable for pins GIOB [7:0]
7-0	GIOENACLR_0	R/W	0h	Interrupt enable for pins GIOA [7:0]

### 17.5.7 GIORLVLSET Register (Offset = 18h) [Reset = 00000000h]

GIORLVLSET is shown in [Table 17-9](#).

Return to the [Summary Table](#).

GPIO high priority interrupt for pins [0:1] GPIO[7:0].

**Table 17-9. GIORLVLSET Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIORLVLSET_3	R/W	0h	GPIO high priority interrupt for pins GIOD [7:0]
23-16	GIORLVLSET_2	R/W	0h	GPIO high priority interrupt for pins GIOC [7:0]
15-8	GIORLVLSET_1	R/W	0h	GPIO high priority interrupt for pins GIOB [7:0]
7-0	GIORLVLSET_0	R/W	0h	GPIO high priority interrupt for pins GIOA [7:0]

### 17.5.8 GIORLCLR Register (Offset = 1Ch) [Reset = 00000000h]

GIORLCLR is shown in [Table 17-10](#).

Return to the [Summary Table](#).

GPIO low priority interrupt for pins [0:1] GPIO[7:0].

**Table 17-10. GIORLCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIORLCLR_3	R/W	0h	GPIO low priority interrupt for pins GIOD [7:0]
23-16	GIORLCLR_2	R/W	0h	GPIO low priority interrupt for pins GIOC [7:0]
15-8	GIORLCLR_1	R/W	0h	GPIO low priority interrupt for pins GIOB [7:0]
7-0	GIORLCLR_0	R/W	0h	GPIO low priority interrupt for pins GIOA [7:0]

### 17.5.9 GIOFLG Register (Offset = 20h) [Reset = 00000000h]

GIOFLG is shown in [Table 17-11](#).

Return to the [Summary Table](#).

GIO flag for pins [0:1] GIO[7:0].

**Table 17-11. GIOFLG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	GIOFLG_3	R/W	0h	GIO flag for pins GIOD [7:0]
23-16	GIOFLG_2	R/W	0h	GIO flag for pins GIOC [7:0]
15-8	GIOFLG_1	R/W	0h	GIO flag for pins GIOB [7:0]
7-0	GIOFLG_0	R/W	0h	GIO flag for pins GIOA [7:0]

### 17.5.10 GIOFFA Register (Offset = 24h) [Reset = 00000000h]

GIOFFA is shown in [Table 17-12](#).

Return to the [Summary Table](#).

Index bits for currently pending high-priority interrupt Register A

**Table 17-12. GIOFFA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	NU1	R/W	0h	Reserved
5-0	GIOFFA	R/W	0h	Index bits for currently pending high-priority interrupt Register A

### 17.5.11 GIOFFB Register (Offset = 28h) [Reset = 00000000h]

GIOFFB is shown in [Table 17-13](#).

Return to the [Summary Table](#).

Index bits for currently pending high-priority interrupt Register B

**Table 17-13. GIOFFB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	NU2	R/W	0h	Reserved
5-0	GIOFFB	R/W	0h	Index bits for currently pending high-priority interrupt Register B



### 17.5.12 GIOEMUA Register (Offset = 2Ch) [Reset = 00000000h]

GIOEMUA is shown in [Table 17-14](#).

Return to the [Summary Table](#).

GIO emulation register A

**Table 17-14. GIOEMUA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	NU3	R/W	0h	Reserved
5-0	GIOEMUA	R/W	0h	GIO emulation register A

### 17.5.13 GIOEMUB Register (Offset = 30h) [Reset = 00000000h]

GIOEMUB is shown in [Table 17-15](#).

Return to the [Summary Table](#).

GIO emulation register B

**Table 17-15. GIOEMUB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-6	NU4	R/W	0h	Reserved
5-0	GIOEMUB	R/W	0h	GIO emulation register B

### 17.5.14 GIODIRA Register (Offset = 34h) [Reset = 00000000h]

GIODIRA is shown in [Table 17-16](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port A

**Table 17-16. GIODIRA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU5	R/W	0h	Reserved
7-0	GIODIRA	R/W	0h	GPIO data direction of pins in Port A

### 17.5.15 GIODINA Register (Offset = 38h) [Reset = 00000000h]

GIODINA is shown in [Table 17-17](#).

Return to the [Summary Table](#).

GIO data input for pins in port A

**Table 17-17. GIODINA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU11	R/W	0h	Reserved
7-0	GIODINA	R/W	0h	GIO data input for pins in port A

### 17.5.16 GIODOUTA Register (Offset = 3Ch) [Reset = 00000000h]

GIODOUTA is shown in [Table 17-18](#).

Return to the [Summary Table](#).

GPIO data output for pins in port A

**Table 17-18. GIODOUTA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU17	R/W	0h	Reserved
7-0	GIODOUTA	R/W	0h	GPIO data output for pins in port A

### 17.5.17 GIOSETA Register (Offset = 40h) [Reset = 00000000h]

GIOSETA is shown in [Table 17-19](#).

Return to the [Summary Table](#).

GIO data set for port A

**Table 17-19. GIOSETA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU23	R/W	0h	Reserved
7-0	GIODSETA	R/W	0h	GIO data set for port A

### 17.5.18 GIOCLRA Register (Offset = 44h) [Reset = 00000000h]

GIOCLRA is shown in [Table 17-20](#).

Return to the [Summary Table](#).

GIO data clear for port A

**Table 17-20. GIOCLRA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU29	R/W	0h	Reserved
7-0	GIODCLRA	R/W	0h	GIO data clear for port A

### 17.5.19 GIOPDRA Register (Offset = 48h) [Reset = 00000000h]

GIOPDRA is shown in [Table 17-21](#).

Return to the [Summary Table](#).

GPIO open drain for port A

**Table 17-21. GIOPDRA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU35	R/W	0h	Reserved
7-0	GIOPDRA	R/W	0h	GPIO open drain for port A



### 17.5.20 GIOPULDISA Register (Offset = 4Ch) [Reset = 00000000h]

GIOPULDISA is shown in [Table 17-22](#).

Return to the [Summary Table](#).

GPIO pull disable for port A

**Table 17-22. GIOPULDISA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU	R/W	0h	Reserved
7-0	GIOPULDISA	R/W	0h	GPIO pull disable for port A

### 17.5.21 GIOPSLA Register (Offset = 50h) [Reset = 00000000h]

GIOPSLA is shown in [Table 17-23](#).

Return to the [Summary Table](#).

GIO pul select for port A

**Table 17-23. GIOPSLA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU35	R/W	0h	Reserved
7-0	GIOPSLA	R/W	0h	GIO pull select for port A

### 17.5.22 GIODIRB Register (Offset = 54h) [Reset = 00000000h]

GIODIRB is shown in [Table 17-24](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port B

**Table 17-24. GIODIRB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU6	R/W	0h	Reserved
7-0	GIODIRB	R/W	0h	GPIO data direction of pins in Port B

### 17.5.23 GIODINB Register (Offset = 58h) [Reset = 00000000h]

GIODINB is shown in [Table 17-25](#).

Return to the [Summary Table](#).

GPIO data input for pins in port B

**Table 17-25. GIODINB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU12	R/W	0h	Reserved
7-0	GIODINB	R/W	0h	GPIO data input for pins in port B

### 17.5.24 GIODOUTB Register (Offset = 5Ch) [Reset = 00000000h]

GIODOUTB is shown in [Table 17-26](#).

Return to the [Summary Table](#).

GPIO data output for pins in port B

**Table 17-26. GIODOUTB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU18	R/W	0h	Reserved
7-0	GIODOUTB	R/W	0h	GPIO data output for pins in port B

### 17.5.25 GIOSETB Register (Offset = 60h) [Reset = 00000000h]

GIOSETB is shown in [Table 17-27](#).

Return to the [Summary Table](#).

GIO data set for port B

**Table 17-27. GIOSETB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU24	R/W	0h	Reserved
7-0	GIOSETB	R/W	0h	GIO data set for port B

### 17.5.26 GIOCLRB Register (Offset = 64h) [Reset = 00000000h]

GIOCLRB is shown in [Table 17-28](#).

Return to the [Summary Table](#).

GIO data clear for port B

**Table 17-28. GIOCLRB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU30	R/W	0h	Reserved
7-0	GIODCLRB	R/W	0h	GIO data clear for port B

### 17.5.27 GIOPDRB Register (Offset = 68h) [Reset = 00000000h]

GIOPDRB is shown in [Table 17-29](#).

Return to the [Summary Table](#).

GPIO open drain for port B

**Table 17-29. GIOPDRB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOPDRB	R/W	0h	GPIO open drain for port B



### 17.5.28 GIOPULDISB Register (Offset = 6Ch) [Reset = 00000000h]

GIOPULDISB is shown in [Table 17-30](#).

Return to the [Summary Table](#).

GPIO pull disable for port B

**Table 17-30. GIOPULDISB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOPULDISB	R/W	0h	GPIO pull disable for port B

### 17.5.29 GIOPSLB Register (Offset = 70h) [Reset = 00000000h]

GIOPSLB is shown in [Table 17-31](#).

Return to the [Summary Table](#).

GPIO pul select for port B

**Table 17-31. GIOPSLB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOPSLB	R/W	0h	GPIO pull select for port B

### 17.5.30 GIODIRC Register (Offset = 74h) [Reset = 00000000h]

GIODIRC is shown in [Table 17-32](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port C

**Table 17-32. GIODIRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU7	R/W	0h	Reserved
7-0	GIODIRC	R/W	0h	GPIO data direction of pins in Port C

### 17.5.31 GIODINC Register (Offset = 78h) [Reset = 00000000h]

GIODINC is shown in [Table 17-33](#).

Return to the [Summary Table](#).

GPIO data input for pins in port C

**Table 17-33. GIODINC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU13	R/W	0h	Reserved
7-0	GIODINC	R/W	0h	GPIO data input for pins in port C

### 17.5.32 GIODOUTC Register (Offset = 7Ch) [Reset = 00000000h]

GIODOUTC is shown in [Table 17-34](#).

Return to the [Summary Table](#).

GPIO data output for pins in port C

**Table 17-34. GIODOUTC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU19	R/W	0h	Reserved
7-0	GIODOUTC	R/W	0h	GPIO data output for pins in port C

### 17.5.33 GIOSETC Register (Offset = 80h) [Reset = 00000000h]

GIOSETC is shown in [Table 17-35](#).

Return to the [Summary Table](#).

GIO data set for port C

**Table 17-35. GIOSETC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU25	R/W	0h	Reserved
7-0	GIOSETC	R/W	0h	GIO data set for port C

### 17.5.34 GIOCLRC Register (Offset = 84h) [Reset = 00000000h]

GIOCLRC is shown in [Table 17-36](#).

Return to the [Summary Table](#).

GIO data clear for port C

**Table 17-36. GIOCLRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU31	R/W	0h	Reserved
7-0	GIODCLRC	R/W	0h	GIO data clear for port C

### 17.5.35 GIOPDRC Register (Offset = 88h) [Reset = 00000000h]

GIOPDRC is shown in [Table 17-37](#).

Return to the [Summary Table](#).

GPIO open drain for port C

**Table 17-37. GIOPDRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOPDRC	R/W	0h	GPIO open drain for port C



### 17.5.36 GIOPULDISC Register (Offset = 8Ch) [Reset = 00000000h]

GIOPULDISC is shown in [Table 17-38](#).

Return to the [Summary Table](#).

GPIO pul disable for port C

**Table 17-38. GIOPULDISC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOPULDISC	R/W	0h	GPIO pull disable for port C

### 17.5.37 GIOPSLC Register (Offset = 90h) [Reset = 00000000h]

GIOPSLC is shown in [Table 17-39](#).

Return to the [Summary Table](#).

GPIO pul select for port C

**Table 17-39. GIOPSLC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOPSLC	R/W	0h	GPIO pull select for port C

### 17.5.38 GIODIRD Register (Offset = 94h) [Reset = 00000000h]

GIODIRD is shown in [Table 17-40](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port D

**Table 17-40. GIODIRD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU8	R/W	0h	Reserved
7-0	GIODIRD	R/W	0h	GPIO data direction of pins in Port D

### 17.5.39 GIODIND Register (Offset = 98h) [Reset = 00000000h]

GIODIND is shown in [Table 17-41](#).

Return to the [Summary Table](#).

GPIO data input for pins in port D

**Table 17-41. GIODIND Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU14	R/W	0h	Reserved
7-0	GIODIND	R/W	0h	GPIO data input for pins in port D

### 17.5.40 GIODOUTD Register (Offset = 9Ch) [Reset = 00000000h]

GIODOUTD is shown in [Table 17-42](#).

Return to the [Summary Table](#).

GPIO data output for pins in port D

**Table 17-42. GIODOUTD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU20	R/W	0h	Reserved
7-0	GIODOUTD	R/W	0h	GPIO data output for pins in port D

### 17.5.41 GIOSETD Register (Offset = A0h) [Reset = 00000000h]

GIOSETD is shown in [Table 17-43](#).

Return to the [Summary Table](#).

GIO data set for port D

**Table 17-43. GIOSETD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU26	R/W	0h	Reserved
7-0	GIOSETD	R/W	0h	GIO data set for port D

### 17.5.42 GIOCLRDR Register (Offset = A4h) [Reset = 00000000h]

GIOCLRDR is shown in [Table 17-44](#).

Return to the [Summary Table](#).

GIO data clear for port D

**Table 17-44. GIOCLRDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU32	R/W	0h	Reserved
7-0	GIOCLRDR	R/W	0h	GIO data clear for port D

### 17.5.43 GIOPDRD Register (Offset = A8h) [Reset = 00000000h]

GIOPDRD is shown in [Table 17-45](#).

Return to the [Summary Table](#).

GPIO open drain for port D

**Table 17-45. GIOPDRD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOPDRD	R/W	0h	GPIO open drain for port D



### 17.5.44 GIOPULDISD Register (Offset = ACh) [Reset = 00000000h]

GIOPULDISD is shown in [Table 17-46](#).

Return to the [Summary Table](#).

GPIO pull disable for port D

**Table 17-46. GIOPULDISD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOPULDISD	R/W	0h	GPIO pull disable for port D

### 17.5.45 GIOPSLD Register (Offset = B0h) [Reset = 00000000h]

GIOPSLD is shown in [Table 17-47](#).

Return to the [Summary Table](#).

GPIO pul select for port D

**Table 17-47. GIOPSLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOPSLD	R/W	0h	GPIO pull select for port D

### 17.5.46 GIODIRE Register (Offset = B4h) [Reset = 00000000h]

GIODIRE is shown in [Table 17-48](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port E

**Table 17-48. GIODIRE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU9	R/W	0h	Reserved
7-0	GIODIRE	R/W	0h	GPIO data direction of pins in Port E

### 17.5.47 GIODINE Register (Offset = B8h) [Reset = 00000000h]

GIODINE is shown in [Table 17-49](#).

Return to the [Summary Table](#).

GPIO data input for pins in port E

**Table 17-49. GIODINE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU15	R/W	0h	Reserved
7-0	GIODINE	R/W	0h	GPIO data input for pins in port E

### 17.5.48 GIODOUTE Register (Offset = BCh) [Reset = 00000000h]

GIODOUTE is shown in [Table 17-50](#).

Return to the [Summary Table](#).

GPIO data output for pins in port E

**Table 17-50. GIODOUTE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU21	R/W	0h	Reserved
7-0	GIODOUTE	R/W	0h	GPIO data output for pins in port E

### 17.5.49 GIOSETTE Register (Offset = C0h) [Reset = 00000000h]

GIOSETTE is shown in [Table 17-51](#).

Return to the [Summary Table](#).

GIO data set for port E

**Table 17-51. GIOSETTE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU27	R/W	0h	Reserved
7-0	GIOSETTE	R/W	0h	GIO data set for port E

### 17.5.50 GIOCLRE Register (Offset = C4h) [Reset = 00000000h]

GIOCLRE is shown in [Table 17-52](#).

Return to the [Summary Table](#).

GIO data clear for port E

**Table 17-52. GIOCLRE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU33	R/W	0h	Reserved
7-0	GIODCLRE	R/W	0h	GIO data clear for port E

### 17.5.51 GIOPDRE Register (Offset = C8h) [Reset = 00000000h]

GIOPDRE is shown in [Table 17-53](#).

Return to the [Summary Table](#).

GPIO open drain for port E

**Table 17-53. GIOPDRE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPDRE	R/W	0h	GPIO open drain for port E



### 17.5.52 GIOPULDISIE Register (Offset = CCh) [Reset = 00000000h]

GIOPULDISIE is shown in [Table 17-54](#).

Return to the [Summary Table](#).

GPIO pul disable for port E

**Table 17-54. GIOPULDISIE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPULDISIE	R/W	0h	GPIO pull disable for port E

### 17.5.53 GIOSPLE Register (Offset = D0h) [Reset = 00000000h]

GIOPSLE is shown in [Table 17-55](#).

Return to the [Summary Table](#).

GPIO pul select for port E

**Table 17-55. GIOSPLE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPSLE	R/W	0h	GPIO pull select for port E

### 17.5.54 GIODIRF Register (Offset = D4h) [Reset = 00000000h]

GIODIRF is shown in [Table 17-56](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port F

**Table 17-56. GIODIRF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU10	R/W	0h	Reserved
7-0	GIODIRF	R/W	0h	GPIO data direction of pins in Port F

### 17.5.55 GIODINF Register (Offset = D8h) [Reset = 00000000h]

GIODINF is shown in [Table 17-57](#).

Return to the [Summary Table](#).

GPIO data input for pins in Port F

**Table 17-57. GIODINF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU16	R/W	0h	Reserved
7-0	GIODINF	R/W	0h	GPIO data input for pins in port F

### 17.5.56 GIODOUTF Register (Offset = DCh) [Reset = 00000000h]

GIODOUTF is shown in [Table 17-58](#).

Return to the [Summary Table](#).

GPIO data output for pins in Port F

**Table 17-58. GIODOUTF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU22	R/W	0h	Reserved
7-0	GIODOUTF	R/W	0h	GPIO data output for pins in port F

### 17.5.57 GIOSETF Register (Offset = E0h) [Reset = 00000000h]

GIOSETF is shown in [Table 17-59](#).

Return to the [Summary Table](#).

GIO data set for Port F

**Table 17-59. GIOSETF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU28	R/W	0h	Reserved
7-0	GIODSETF	R/W	0h	GIO data set for port F

### 17.5.58 GIOCLRFR Register (Offset = E4h) [Reset = 00000000h]

GIOCLRFR is shown in [Table 17-60](#).

Return to the [Summary Table](#).

GIO data clear for Port F

**Table 17-60. GIOCLRFR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU34	R/W	0h	Reserved
7-0	GIOCLRFR	R/W	0h	GIO data clear for port F

### 17.5.59 GIOPDRF Register (Offset = E8h) [Reset = 00000000h]

GIOPDRF is shown in [Table 17-61](#).

Return to the [Summary Table](#).

GPIO open drain for Port F

**Table 17-61. GIOPDRF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPDRF	R/W	0h	GPIO open drain for port F



### 17.5.60 GIOPULDISF Register (Offset = ECh) [Reset = 00000000h]

GIOPULDISF is shown in [Table 17-62](#).

Return to the [Summary Table](#).

GPIO pull disable for port F

**Table 17-62. GIOPULDISF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPULDISF	R/W	0h	GPIO pull disable for port F

### 17.5.61 GIOPSLF Register (Offset = F0h) [Reset = 00000000h]

GIOPSLF is shown in [Table 17-63](#).

Return to the [Summary Table](#).

GIO pul select for port F

**Table 17-63. GIOPSLF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPSLF	R/W	0h	GIO pull select for port F

### 17.5.62 GIODIRG Register (Offset = F4h) [Reset = 00000000h]

GIODIRG is shown in [Table 17-64](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port G

**Table 17-64. GIODIRG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU9	R/W	0h	Reserved
7-0	GIODIRG	R/W	0h	GPIO data direction of pins in Port G

### 17.5.63 GIODING Register (Offset = F8h) [Reset = 00000000h]

GIODING is shown in [Table 17-65](#).

Return to the [Summary Table](#).

GPIO data input for pins in port G

**Table 17-65. GIODING Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU15	R/W	0h	Reserved
7-0	GIODING	R/W	0h	GPIO data input for pins in port G

### 17.5.64 GIODOUTG Register (Offset = FCh) [Reset = 00000000h]

GIODOUTG is shown in [Table 17-66](#).

Return to the [Summary Table](#).

GPIO data output for pins in port G

**Table 17-66. GIODOUTG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU21	R/W	0h	Reserved
7-0	GIODOUTG	R/W	0h	GPIO data output for pins in port G

### 17.5.65 GIOSETG Register (Offset = 100h) [Reset = 00000000h]

GIOSETG is shown in [Table 17-67](#).

Return to the [Summary Table](#).

GIO data set for port G

**Table 17-67. GIOSETG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU27	R/W	0h	Reserved
7-0	GIODSETG	R/W	0h	GIO data set for port G

### 17.5.66 GIOCLRG Register (Offset = 104h) [Reset = 00000000h]

GIOCLRG is shown in [Table 17-68](#).

Return to the [Summary Table](#).

GIO data clear for port G

**Table 17-68. GIOCLRG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU33	R/W	0h	Reserved
7-0	GIODCLRG	R/W	0h	GIO data clear for port G

### 17.5.67 GIOPDRG Register (Offset = 108h) [Reset = 00000000h]

GIOPDRG is shown in [Table 17-69](#).

Return to the [Summary Table](#).

GPIO open drain for port G

**Table 17-69. GIOPDRG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPDRG	R/W	0h	GPIO open drain for port G



### 17.5.68 GIOPULDISG Register (Offset = 10Ch) [Reset = 00000000h]

GIOPULDISG is shown in [Table 17-70](#).

Return to the [Summary Table](#).

GPIO pull disable for port G

**Table 17-70. GIOPULDISG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPULDISG	R/W	0h	GPIO pull disable for port G

### 17.5.69 GIOPSLG Register (Offset = 110h) [Reset = 00000000h]

GIOPSLG is shown in [Table 17-71](#).

Return to the [Summary Table](#).

GPIO pul select for port G

**Table 17-71. GIOPSLG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOPSLG	R/W	0h	GPIO pull select for port G

### 17.5.70 GIODIRH Register (Offset = 114h) [Reset = 00000000h]

GIODIRH is shown in [Table 17-72](#).

Return to the [Summary Table](#).

GPIO data direction of pins in Port H

**Table 17-72. GIODIRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU10	R/W	0h	Reserved
7-0	GIODIRH	R/W	0h	GPIO data direction of pins in Port H

### 17.5.71 GIODINH Register (Offset = 118h) [Reset = 00000000h]

GIODINH is shown in [Table 17-73](#).

Return to the [Summary Table](#).

GPIO data input for pins in Port H

**Table 17-73. GIODINH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU16	R/W	0h	Reserved
7-0	GIODINH	R/W	0h	GPIO data input for pins in port H

### 17.5.72 GIODOUTH Register (Offset = 11Ch) [Reset = 00000000h]

GIODOUTH is shown in [Table 17-74](#).

Return to the [Summary Table](#).

GPIO data output for pins in Port H

**Table 17-74. GIODOUTH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU22	R/W	0h	Reserved
7-0	GIODOUTH	R/W	0h	GPIO data output for pins in port H

### 17.5.73 GIOSETH Register (Offset = 120h) [Reset = 00000000h]

GIOSETH is shown in [Table 17-75](#).

Return to the [Summary Table](#).

GIO data set for Port H

**Table 17-75. GIOSETH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU28	R/W	0h	Reserved
7-0	GIOSETH	R/W	0h	GIO data set for port H

### 17.5.74 GIOCLRH Register (Offset = 124h) [Reset = 00000000h]

GIOCLRH is shown in [Table 17-76](#).

Return to the [Summary Table](#).

GIO data clear for Port H

**Table 17-76. GIOCLRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU34	R/W	0h	Reserved
7-0	GIODCLRH	R/W	0h	GIO data clear for port H

### 17.5.75 GIOPDRH Register (Offset = 128h) [Reset = 00000000h]

GIOPDRH is shown in [Table 17-77](#).

Return to the [Summary Table](#).

GPIO open drain for Port H

**Table 17-77. GIOPDRH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPDRH	R/W	0h	GPIO open drain for port H



### 17.5.76 GIOPULDISH Register (Offset = 12Ch) [Reset = 00000000h]

GIOPULDISH is shown in [Table 17-78](#).

Return to the [Summary Table](#).

GPIO pull disable for port H

**Table 17-78. GIOPULDISH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPULDISH	R/W	0h	GPIO pull disable for port H

### 17.5.77 GIOPSLH Register (Offset = 130h) [Reset = 00000000h]

GIOPSLH is shown in [Table 17-79](#).

Return to the [Summary Table](#).

GPIO pul select for port H

**Table 17-79. GIOPSLH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOPSLH	R/W	0h	GPIO pull select for port H

### 17.5.78 GIOSRCA Register (Offset = 134h) [Reset = 00000000h]

GIOSRCA is shown in [Table 17-80](#).

Return to the [Summary Table](#).

GPIO slew rate select for port A

**Table 17-80. GIOSRCA Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU35	R/W	0h	Reserved
7-0	GIOSRCA	R/W	0h	GPIO slew rate control for port A

### 17.5.79 GIOSRCB Register (Offset = 138h) [Reset = 00000000h]

GIOSRCB is shown in [Table 17-81](#).

Return to the [Summary Table](#).

GPIO slew rate select for port B

**Table 17-81. GIOSRCB Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU36	R/W	0h	Reserved
7-0	GIOSRCB	R/W	0h	GPIO slew rate control for port B

### 17.5.80 GIOSRCC Register (Offset = 13Ch) [Reset = 00000000h]

GIOSRCC is shown in [Table 17-82](#).

Return to the [Summary Table](#).

GPIO slew rate select for port C

**Table 17-82. GIOSRCC Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU37	R/W	0h	Reserved
7-0	GIOSRCC	R/W	0h	GPIO slew rate control for port C

### 17.5.81 GIOSRCD Register (Offset = 140h) [Reset = 00000000h]

GIOSRCD is shown in [Table 17-83](#).

Return to the [Summary Table](#).

GPIO slew rate select for port D

**Table 17-83. GIOSRCD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU38	R/W	0h	Reserved
7-0	GIOSRCD	R/W	0h	GPIO slew rate control for port D

### 17.5.82 GIOSRCE Register (Offset = 144h) [Reset = 00000000h]

GIOSRCE is shown in [Table 17-84](#).

Return to the [Summary Table](#).

GPIO slew rate select for port E

**Table 17-84. GIOSRCE Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOSRCE	R/W	0h	GPIO slew rate control for port E

### 17.5.83 GIOSRCF Register (Offset = 148h) [Reset = 00000000h]

GIOSRCF is shown in [Table 17-85](#).

Return to the [Summary Table](#).

GPIO slew rate select for port F

**Table 17-85. GIOSRCF Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOSRCF	R/W	0h	GPIO slew rate control for port F



### 17.5.84 GIOSRCG Register (Offset = 14Ch) [Reset = 00000000h]

GIOSRCG is shown in [Table 17-86](#).

Return to the [Summary Table](#).

GPIO slew rate select for port G

**Table 17-86. GIOSRCG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU39	R/W	0h	Reserved
7-0	GIOSRCG	R/W	0h	GPIO slew rate control for port G

### 17.5.85 GIOSRCH Register (Offset = 150h) [Reset = 00000000h]

GIOSRCH is shown in [Table 17-87](#).

Return to the [Summary Table](#).

GIO slew rate select for port H

**Table 17-87. GIOSRCH Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	NU40	R/W	0h	Reserved
7-0	GIOSRCH	R/W	0h	GIO slew rate control for port H

## 17.6 I/O Control Summary

The behavior of the output buffer and the pull control is summarized in [Table 17-88](#).

**Table 17-88. Output Buffer and Pull Control Behavior for GPIO Pins**

Module under Reset?	Pin Direction (GIODIR) <sup>(1) (2)</sup>	Open Drain Enable (GIOPDR) <sup>(1)</sup>	Pull Disable (GIOPULDIS) <sup>(1) (3)</sup>	Pull Select (GIOPSL) <sup>(1) (4)</sup>	Pull Control	Output Buffer <sup>(5)</sup>
Yes	X	X	X	X	Enabled	Disabled
No	0	X	0	0	Pull down	Disabled
No	0	X	0	1	Pull up	Disabled
No	0	X	1	0	Disabled	Disabled
No	0	X	1	1	Disabled	Disabled
No	1	0	X	X	Disabled	Enabled
No	1	1	X	X	Disabled	Enabled

(1) X = Don't care

(2) GIODIR = 0 for input; = 1 for output

(3) GIOPULDIS = 0 for enabling pull control; = 1 for disabling pull control

(4) GIOPSL = 0 for pull-down functionality; = 1 for pull-up functionality

(5) If open drain is enabled, output buffer will be disabled if a high level (1) is being output.

## Chapter 18

# Enhanced Pulse Width Modulator (ePWM) Module

---



The enhanced pulse width modulator (ePWM) peripheral is a key element in controlling many of the power electronic systems found in both commercial and industrial equipments. The features supported by the ePWM make it especially suitable for digital motor control.

<b>18.1 Introduction.....</b>	<b><a href="#">4695</a></b>
<b>18.2 ePWM Submodules.....</b>	<b><a href="#">4699</a></b>
<b>18.3 Application Examples.....</b>	<b><a href="#">4759</a></b>
<b>18.4 ePWM Module Control and Status Registers.....</b>	<b><a href="#">4774</a></b>

## 18.1 Introduction

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention. It needs to be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, the ePWM is built up from smaller single channel modules with separate resources that can operate together as required to form a system. This modular approach results in an orthogonal architecture and provides a more transparent view of the peripheral structure, helping users to understand its operation quickly.

In this document the letter x within a signal or module name is used to indicate a generic ePWM instance on a device. For example, output signals EPWMxA and EPWMxB refer to the output signals from the ePWMx instance. Thus, EPWM1A and EPWM1B belong to ePWM1 and likewise EPWM4A and EPWM4B belong to ePWM4.

### 18.1.1 Submodule Overview

The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWMxA and EPWMxB. Multiple ePWM modules are instantiated within a device as shown in [Figure 18-1](#). Each ePWM instance is identical and is indicated by a numerical value starting with 1. For example, ePWM1 is the first instance and ePWM3 is the third instance in the system and ePWMx indicates any instance.

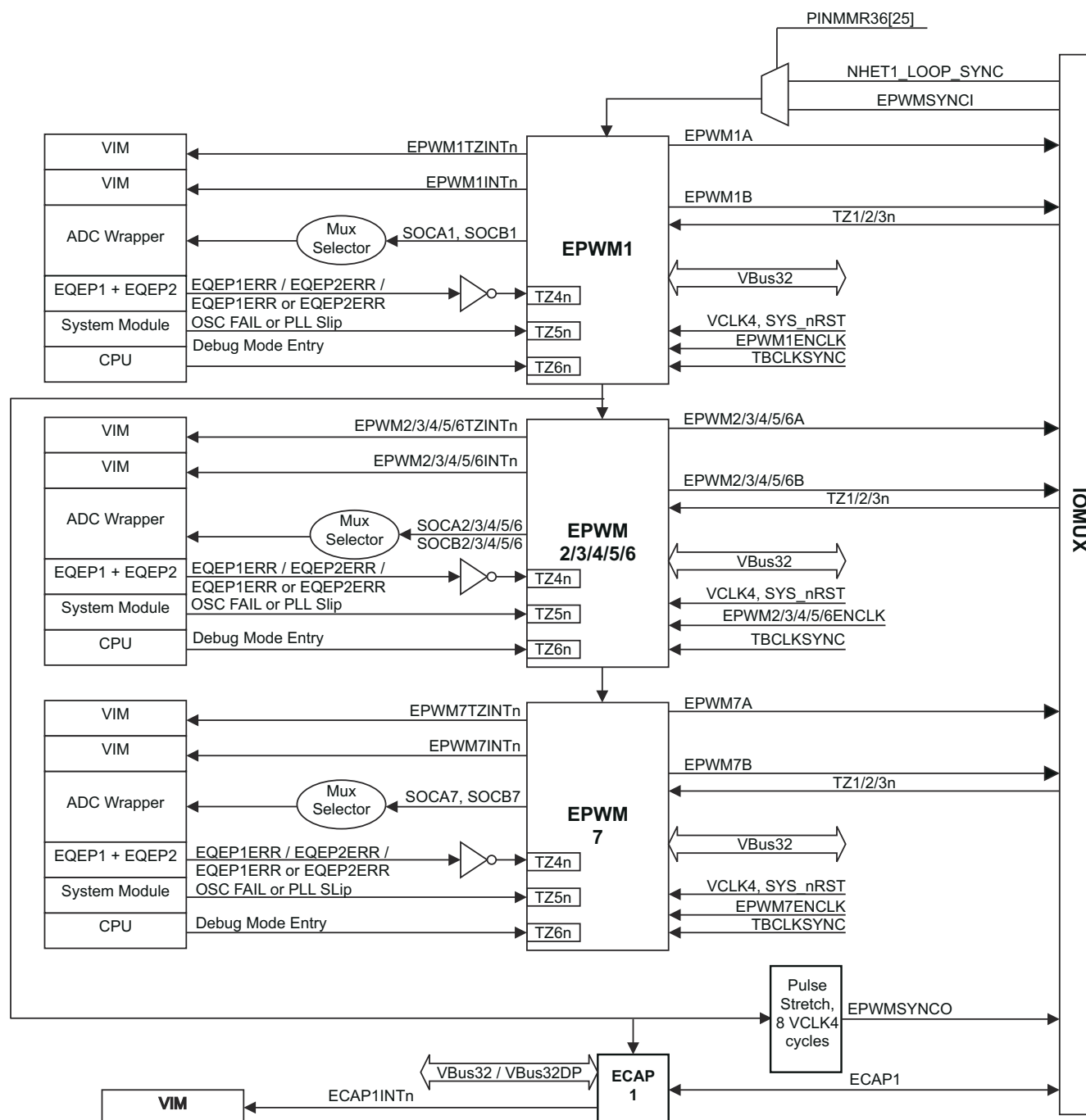
The ePWM modules are chained together via a clock synchronization scheme that allows them to operate as a single system when required. Additionally, this synchronization scheme can be extended to the capture peripheral modules (eCAP). Modules can also operate stand-alone.

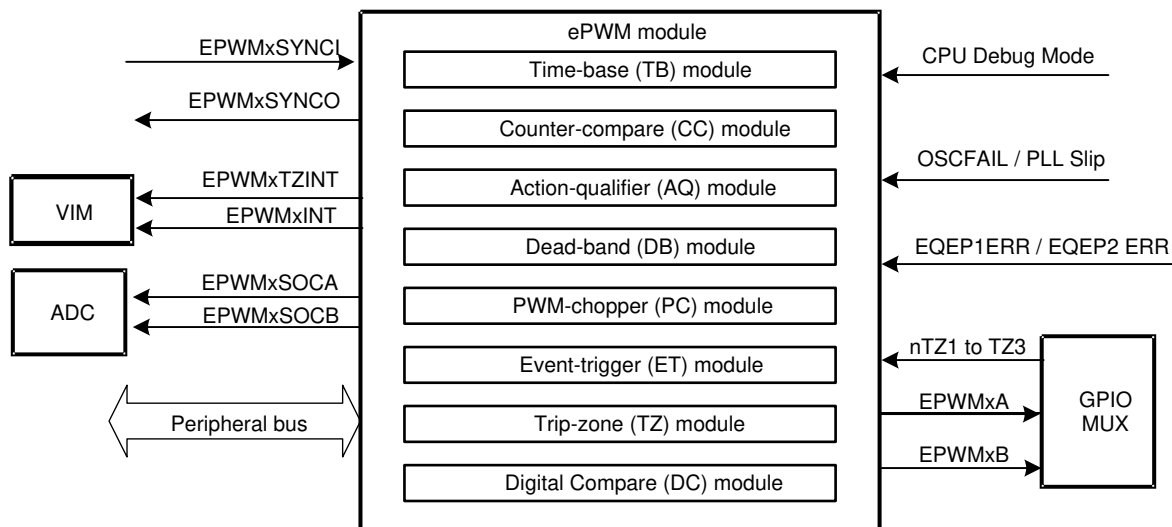
Each ePWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Two PWM outputs (EPWMxA and EPWMxB) that can be used in the following configurations:
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Programmable phase-control support for lag or lead operation relative to other ePWM modules.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- All events can trigger both CPU interrupts and ADC start of conversion (SOC)
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

Each ePWM module is connected to the input/output signals shown in [Figure 18-1](#). The signals are described in detail in subsequent sections.

Each ePWM module consists of eight submodules and is connected within a system via the signals shown in [Figure 18-2](#).


**Figure 18-1. Multiple ePWM Modules**



**Figure 18-2. Submodules and Signal Connections for an ePWM Module**

The main signals used by the ePWM module are:

- **PWM output signals (EPWMxA and EPWMxB).**

The PWM output signals are made available external to the device through the Pinmux.

- **Trip-zone signals (TZ1 to TZ6).**

These input signals alert the ePWM module of fault conditions external to the ePWM module. Each ePWM module can be configured to either use or ignore any of the trip-zone signals. The TZ1 to TZ3 trip-zone signals can be configured as asynchronous inputs, or double-synchronized using VCLK4, or double-synchronized and filtered through a 6-VCLK4-cycle counter before connecting to the ePWM modules. This selection is done by configuring registers in the Pinmux. TZ4 is connected to an inverted eQEP1 error signal (EQEP1ERR), or to an inverted eQEP2 error signal (EQEP2ERR), or an OR-combination of EQEP1ERR and EQEP2ERR. This selection is also done via the Pinmux registers. TZ5 is connected to the system clock fail status. This is asserted whenever an oscillator failure is detected, or a PLL slip is detected. TZ6 is connected to the debug mode entry indicator output from the CPU. This allows you to configure a trip action when the CPU halts.

- **Time-base synchronization input (EPWMxSYNCl) and output (EPWMxSYNCO) signals.**

The synchronization signals daisy chain the ePWM modules together. Each module can be configured to either use or ignore its synchronization input. The clock synchronization input and output signal are brought out to pins only for ePWM1 (ePWM module #1). The synchronization output for ePWM1 (EPWM1SYNCO) is also connected to the SYNCl of the first enhanced capture module (eCAP1).

- **ADC start-of-conversion signals (EPWMxSOCA and EPWMxSOCB).**

Each ePWM module has two ADC start of conversion signals. Any ePWM module can trigger a start of conversion. Which event triggers the start of conversion is configured in the Event-Trigger submodule of the ePWM.

- **Peripheral Bus**

The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

## 18.1.2 Register Mapping

The complete ePWM module control and status register set is grouped by submodule as shown in [Table 18-1](#). Each register set is duplicated for each instance of the ePWM module. The start address for each ePWM register file instance on a device is specified in the specific part's datasheet.

**Table 18-1. ePWM Module Control and Status Register Set Grouped by Submodule**

Name	Address Offset <sup>(1)</sup>	Size (×16)	Shadow	Privileged Mode Write Only?	Description
<b>Time-Base Submodule Registers</b>					
TBCTL	0x0002	1	No	No	Time-Base Control Register
TBSTS	0x0000	1	No	No	Time-Base Status Register
Reserved	0x0006	1	–	–	Reserved
TBPHS	0x0004	1	No	No	Time-Base Phase Register
TBCTR	0x000A	1	No	No	Time-Base Counter Register
TBPRD	0x0008	1	Yes	No	Time-Base Period Register
Reserved	0x000E	1	–	–	Reserved
<b>Counter-Compare Submodule Registers</b>					
CMPCTL	0x000C	1	No	No	Counter-Compare Control Register
Reserved	0x0012	1	–	–	Reserved
CMPA	0x0010	1	Yes	No	Counter-Compare A Register
CMPB	0x0016	1	Yes	No	Counter-Compare B Register
<b>Action-Qualifier Submodule Registers</b>					
AQCTLA	0x0014	1	No	No	Action-Qualifier Control Register for Output A (EPWMxA)
AQCTLB	0x001A	1	No	No	Action-Qualifier Control Register for Output B (EPWMxB)
AQSFRC	0x0018	1	No	No	Action-Qualifier Software Force Register
AQCSFRC	0x001E	1	Yes	No	Action-Qualifier Continuous S/W Force Register Set
<b>Dead-Band Generator Submodule Registers</b>					
DBCTL	0x001C	1	No	No	Dead-Band Generator Control Register
DBRED	0x0022	1	No	No	Dead-Band Generator Rising Edge Delay Count Register
DBFED	0x0020	1	No	No	Dead-Band Generator Falling Edge Delay Count Register
<b>Trip-Zone Submodule Registers</b>					
TZSEL	0x0026	1	No	Yes	Trip-Zone Select Register
TZDCSEL	0x0024	1	No	Yes	Trip Zone Digital Compare Select Register
TZCTL	0x002A	1	No	Yes	Trip-Zone Control Register
TZEINT	0x0028	1	No	Yes	Trip-Zone Enable Interrupt Register
TZFLG	0x002E	1	No	No	Trip-Zone Flag Register
TZCLR	0x002C	1	No	Yes	Trip-Zone Clear Register
TZFRC	0x0032	1	No	Yes	Trip-Zone Force Register
<b>Event-Trigger Submodule Registers</b>					
ETSEL	0x0030	1	No	No	Event-Trigger Selection Register
ETPS	0x0036	1	No	No	Event-Trigger Pre-Scale Register
ETFLG	0x0034	1	No	No	Event-Trigger Flag Register
ETCLR	0x003A	1	No	No	Event-Trigger Clear Register
ETFRC	0x0038	1	No	No	Event-Trigger Force Register
<b>PWM-Chopper Submodule Registers</b>					
PCCTL	0x003E	1	No	No	PWM-Chopper Control Register



**Table 18-1. ePWM Module Control and Status Register Set Grouped by Submodule (continued)**

Name	Address Offset <sup>(1)</sup>	Size (×16)	Shadow	Privileged Mode Write Only?	Description
<b>Digital Compare Event Registers</b>					
DCTRISEL	0x0062	1	No	Yes	Digital Compare Trip Select Register
DCACTL	0x0060	1	No	Yes	Digital Compare A Control Register
DCBCTL	0x0066	1	No	Yes	Digital Compare B Control Register
DCFCTL	0x0064	1	No	Yes	Digital Compare Filter Control Register
DCCAPCTL	0x006A	1	No	Yes	Digital Compare Capture Control Register
DCFOFFSET	0x0068	1	Writes	No	Digital Compare Filter Offset Register
DCFOFFSETCNT	0x006E	1	No	No	Digital Compare Filter Offset Counter Register
DCFWINDOW	0x006C	1	No	No	Digital Compare Filter Window Register
DCFWINDOWCNT	0x0072	1	No	No	Digital Compare Filter Window Counter Register
DCCAP	0x0070	1	Yes	No	Digital Compare Counter Capture Register

(1) Locations not shown are reserved.

## 18.2 ePWM Submodules

Eight submodules are included in every ePWM peripheral. Each of these submodules performs specific tasks that can be configured by software.

### 18.2.1 Overview

[Table 18-2](#) lists the eight key submodules together with a list of their main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, then you should see the counter-compare submodule in [Section 18.2.3](#) for relevant details.

**Table 18-2. Submodule Configuration Parameters**

Submodule	Configuration Parameter or Option
Time-base (TB)	<ul style="list-style-type: none"> <li>Scale the time-base clock (TBCLK) relative to the system clock (VCLK4).</li> <li>Configure the PWM time-base counter (TBCTR) frequency or period.</li> <li>Set the mode for the time-base counter: <ul style="list-style-type: none"> <li>count-up mode: used for asymmetric PWM</li> <li>count-down mode: used for asymmetric PWM</li> <li>count-up-and-down mode: used for symmetric PWM</li> </ul> </li> <li>Configure the time-base phase relative to another ePWM module.</li> <li>Synchronize the time-base counter between modules through hardware or software.</li> <li>Configure the direction (up or down) of the time-base counter after a synchronization event.</li> <li>Configure how the time-base counter will behave when the device is halted by an emulator.</li> <li>Specify the source for the synchronization output of the ePWM module: <ul style="list-style-type: none"> <li>Synchronization input signal</li> <li>Time-base counter equal to zero</li> <li>Time-base counter equal to counter-compare B (CMPB)</li> <li>No output synchronization signal generated.</li> </ul> </li> </ul>
Counter-compare (CC)	<ul style="list-style-type: none"> <li>Specify the PWM duty cycle for output EPWMxA and/or output EPWMxB</li> <li>Specify the time at which switching events occur on the EPWMxA or EPWMxB output</li> </ul>

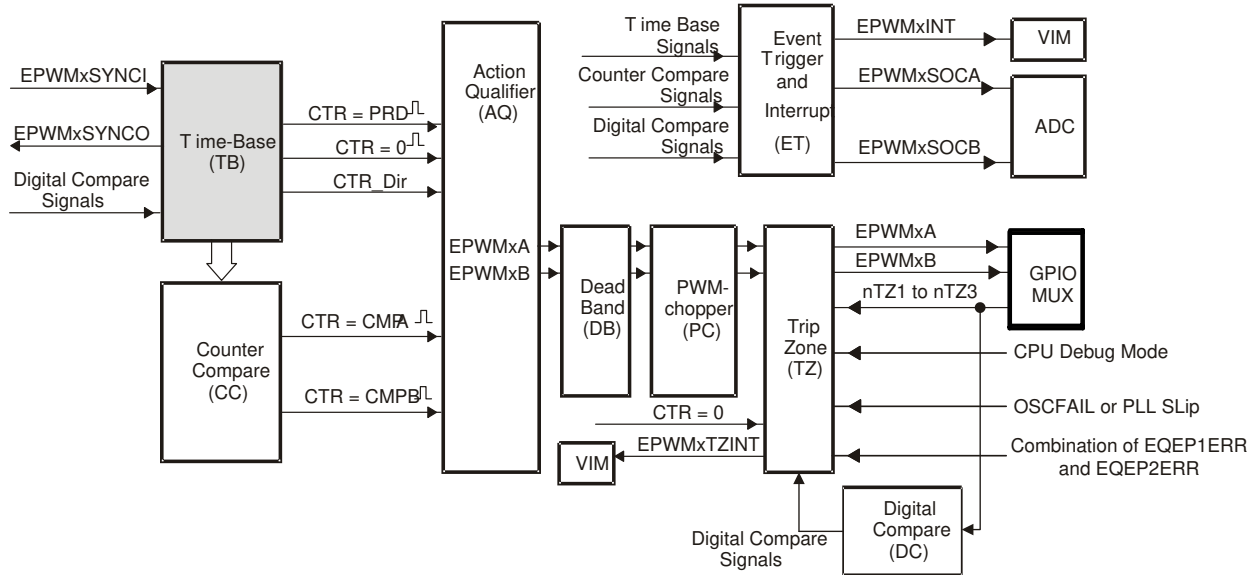
**Table 18-2. Submodule Configuration Parameters (continued)**

Submodule	Configuration Parameter or Option
Action-qualifier (AQ)	<ul style="list-style-type: none"> <li>Specify the type of action taken when a time-base or counter-compare submodule event occurs: <ul style="list-style-type: none"> <li>No action taken</li> <li>Output EPWMxA and/or EPWMxB switched high</li> <li>Output EPWMxA and/or EPWMxB switched low</li> <li>Output EPWMxA and/or EPWMxB toggled</li> </ul> </li> <li>Force the PWM output state through software control</li> <li>Configure and control the PWM dead-band through software</li> </ul>
Dead-band (DB)	<ul style="list-style-type: none"> <li>Control of traditional complementary dead-band relationship between upper and lower switches</li> <li>Specify the output rising-edge-delay value</li> <li>Specify the output falling-edge delay value</li> <li>Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification.</li> <li>Option to enable half-cycle clocking for double resolution.</li> </ul>
PWM-chopper (PC)	<ul style="list-style-type: none"> <li>Create a chopping (carrier) frequency.</li> <li>Pulse width of the first pulse in the chopped pulse train.</li> <li>Duty cycle of the second and subsequent pulses.</li> <li>Bypass the PWM-chopper module entirely. In this case the PWM waveform is passed through without modification.</li> </ul>
Trip-zone (TZ)	<ul style="list-style-type: none"> <li>Configure the ePWM module to react to one, all, or none of the trip-zone signals or digital compare events.</li> <li>Specify the tripping action taken when a fault occurs: <ul style="list-style-type: none"> <li>Force EPWMxA and/or EPWMxB high</li> <li>Force EPWMxA and/or EPWMxB low</li> <li>Force EPWMxA and/or EPWMxB to a high-impedance state</li> <li>Configure EPWMxA and/or EPWMxB to ignore any trip condition.</li> </ul> </li> <li>Configure how often the ePWM will react to each trip-zone signal: <ul style="list-style-type: none"> <li>One-shot</li> <li>Cycle-by-cycle</li> </ul> </li> <li>Enable the trip-zone to initiate an interrupt.</li> <li>Bypass the trip-zone module entirely.</li> </ul>
Event-trigger (ET)	<ul style="list-style-type: none"> <li>Enable the ePWM events that will trigger an interrupt.</li> <li>Enable ePWM events that will trigger an ADC start-of-conversion event.</li> <li>Specify the rate at which events cause triggers (every occurrence or every second or third occurrence)</li> <li>Poll, set, or clear event flags</li> </ul>
Digital-compare (DC)	<ul style="list-style-type: none"> <li>Enables trip zone signals to create events and filtered events</li> <li>Specify event-filtering options to capture TBCTR counter or generate blanking window</li> </ul>

Code examples are provided in the remainder of this document that show how to implement various ePWM module configurations. These examples use the constant definitions in the device *EPwm\_defines.h* file in the device-specific header file and peripheral examples software package.

### 18.2.2 Time-Base (TB) Submodule

Each ePWM module has its own time-base submodule that determines all of the event timing for the ePWM module. Built-in synchronization logic allows the time-base of multiple ePWM modules to work together as a single system. Figure 18-3 illustrates the time-base module's place within the ePWM.



**Figure 18-3. Time-Base Submodule Block Diagram**

#### 18.2.2.1 Purpose of the Time-Base Submodule

You can configure the time-base submodule for the following:

- Specify the ePWM time-base counter (TBCTR) frequency or period to control how often events occur.
- Manage time-base synchronization with other ePWM modules.
- Maintain a phase relationship with other ePWM modules.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - CTR = PRD: Time-base counter equal to the specified period (TBCTR = TBPRD) .
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000).
- Configure the rate of the time-base clock; a prescaled version of the device peripheral clock domain (VCLK4). This allows the time-base counter to increment/decrement at a slower rate.

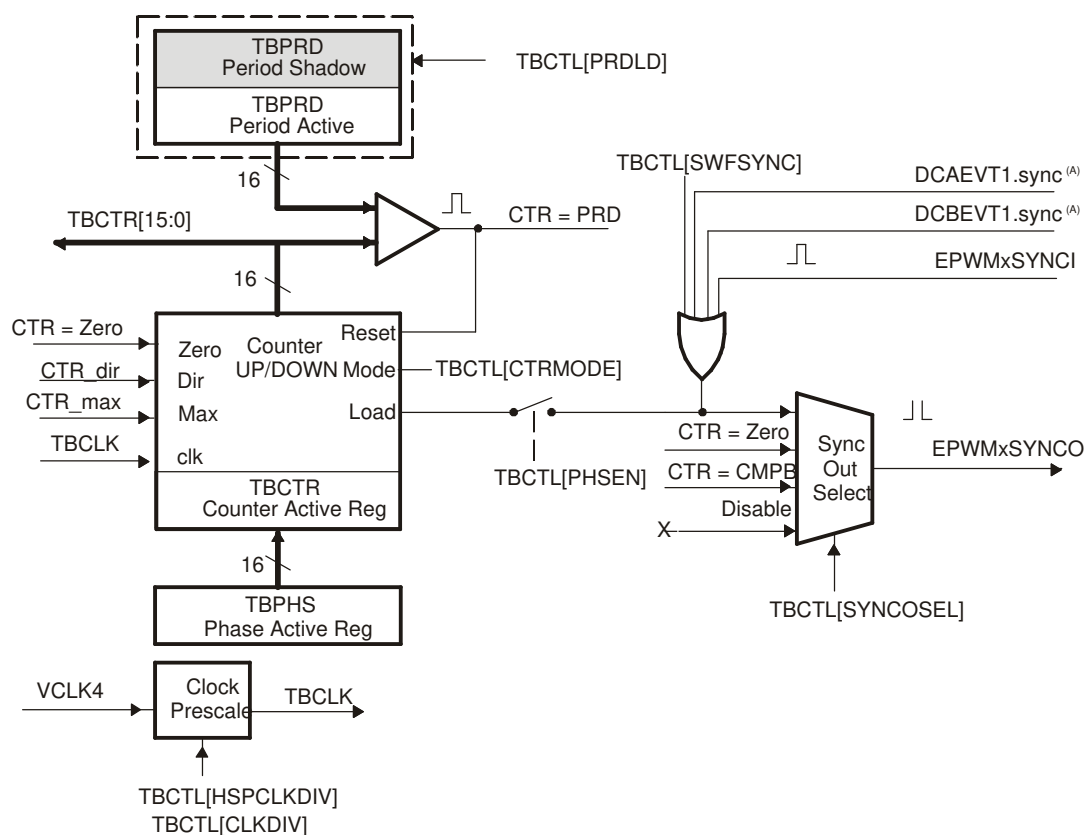
### 18.2.2.2 Controlling and Monitoring the Time-base Submodule

Table 18-3 shows the registers used to control and monitor the time-base submodule.

**Table 18-3. Time-Base Submodule Registers**

Register Name	Address Offset	Shadowed	Description
TBCTL	0x0002	No	Time-Base Control Register
TBSTS	0x0000	No	Time-Base Status Register
TBPHS	0x0004	No	Time-Base Phase Register
TBCTR	0x000A	No	Time-Base Counter Register
TBPRD	0x0008	Yes	Time-Base Period Register

The block diagram in Figure 18-4 shows the critical signals and registers of the time-base submodule. Table 18-4 provides descriptions of the key signals associated with the time-base submodule.



A. These signals are generated by the digital compare (DC) submodule.

**Figure 18-4. Time-Base Submodule Signals and Registers**

**Table 18-4. Key Time-Base Signals**

Signal	Description
EPWMxSYNCl	Time-base synchronization input.  Input pulse used to synchronize the time-base counter with the counter of ePWM module earlier in the synchronization chain. An ePWM peripheral can be configured to use or ignore this signal. For the first ePWM module (EPWM1), this signal comes from a device pin or from the N2HET1 module. For subsequent ePWM modules, this signal is passed from another ePWM peripheral. For example, EPWM2SYNCl is generated by the ePWM1 peripheral, EPWM3SYNCl is generated by ePWM2 and so forth. See <a href="#">Section 18.2.2.3.3</a> for information on the synchronization order of a particular device.
EPWMxSYNCO	Time-base synchronization output.  This output pulse is used to synchronize the counter of an ePWM module later in the synchronization chain. The ePWM module generates this signal from one of three event sources: <ol style="list-style-type: none"> <li>1. EPWMxSYNCl (Synchronization input pulse)</li> <li>2. CTR = Zero: The time-base counter equal to zero (TBCTR = 0x0000).</li> <li>3. CTR = CMPB: The time-base counter equal to the counter-compare B (TBCTR = CMPB) register.</li> </ol>
CTR = PRD	Time-base counter equal to the specified period.  This signal is generated whenever the counter value is equal to the active period register value. That is when TBCTR = TBPRD.
CTR = Zero	Time-base counter equal to zero  This signal is generated whenever the counter value is zero. That is when TBCTR equals 0x0000.
CTR = CMPB	Time-base counter equal to active counter-compare B register (TBCTR = CMPB).  This event is generated by the counter-compare submodule and used by the synchronization out logic
CTR_dir	Time-base counter direction.  Indicates the current direction of the ePWM's time-base counter. This signal is high when the counter is increasing and low when it is decreasing.
CTR_max	Time-base counter equal max value. (TBCTR = 0xFFFF)  Generated event when the TBCTR value reaches its maximum value. This signal is only used only as a status bit
TBCLK	Time-base clock.  This is a prescaled version of the system clock (VCLK4) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

### 18.2.2.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (TBPRD) register and the mode of the time-base counter. [Figure 18-5](#) shows the period ( $T_{pwm}$ ) and frequency ( $F_{pwm}$ ) relationships for the up-count, down-count, and up-down-count time-base counter modes when the period is set to 4 (TBPRD = 4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the system clock (VCLK4).

The time-base counter has three modes of operation selected by the time-base control register (TBCTL):

- **Up-Down-Count Mode:**

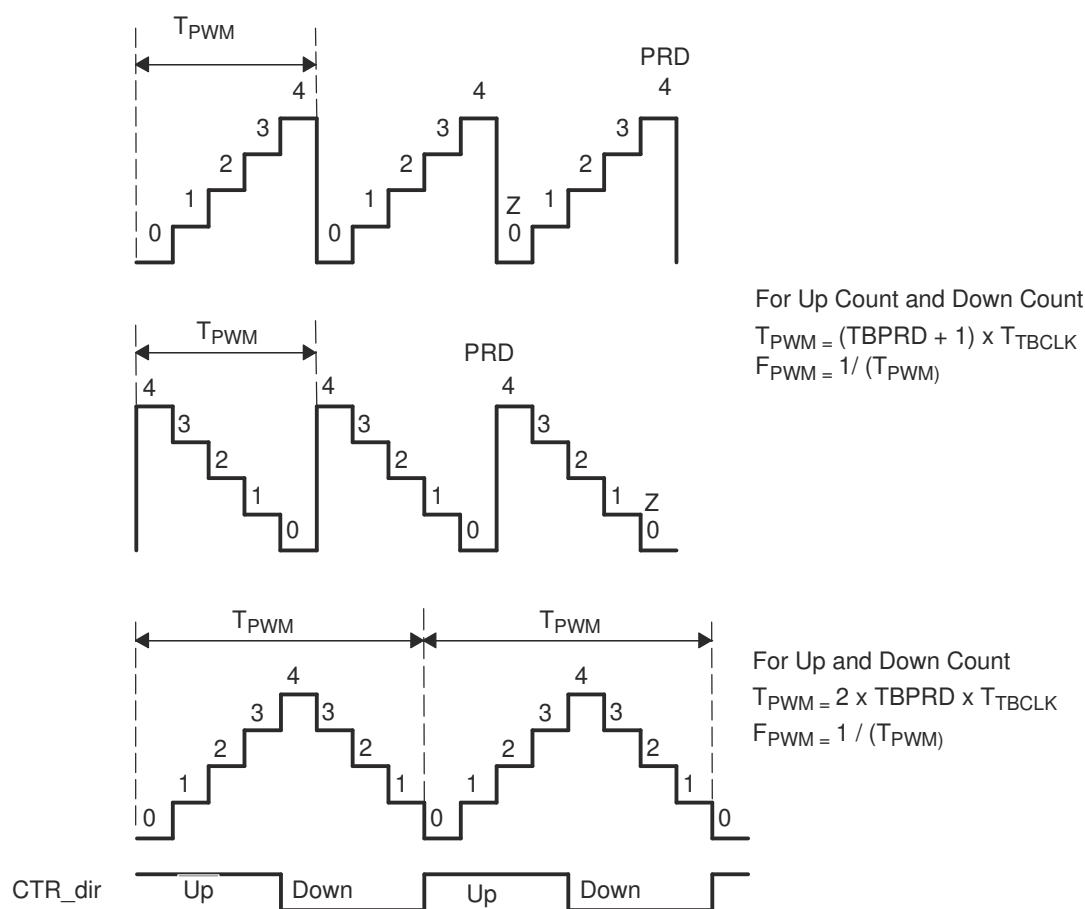
In up-down-count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until it reaches zero. At this point the counter repeats the pattern and begins to increment.

- **Up-Count Mode:**

In this mode, the time-base counter starts from zero and increments until it reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.

- **Down-Count Mode:**

In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until it reaches zero. When it reaches zero, the time-base counter is reset to the period value and it begins to decrement once again.



**Figure 18-5. Time-Base Frequency and Period**

#### 18.2.2.3.1 Time-Base Period Shadow Register

The time-base period register (TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

- **Active Register**

The active register controls the hardware and is responsible for actions that the hardware causes or invokes.

- **Shadow Register**

The shadow register buffers or provides a temporary holding location for the active register. It has no direct effect on any control hardware. At a strategic point in time the shadow register's content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the TBCTL[PRDL] bit. This bit enables and disables the TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:**

The TBPRD shadow register is enabled when TBCTL[PRDL] = 0. Reads from and writes to the TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (TBPRD (Active) ← TBPRD (shadow)) when the time-base counter equals zero (TBCTR = 0x0000). By default the TBPRD shadow register is enabled.

- **Time-Base Period Immediate Load Mode:**

If immediate load mode is selected (TBCTL[PRDL] = 1), then a read from or a write to the TBPRD memory address goes directly to the active register.

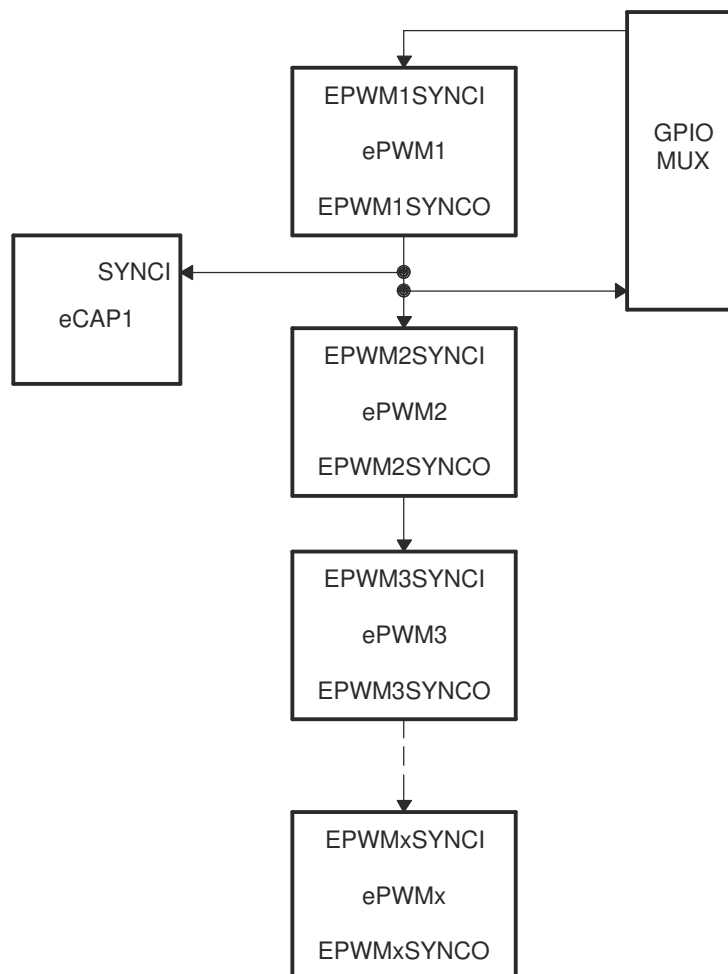
#### 18.2.2.3.2 Time-Base Clock Synchronization

Bit 1 of the device-level multiplexing control module (Pinmux) register PINMMR37 is defined as the TBCLKSYNC bit. The TBCLKSYNC bit allows users to globally synchronize all enabled ePWM modules to the time-base clock (TBCLK). When set, all enabled ePWM module clocks are started with the first rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescalers for each ePWM module must be set identically.

The proper procedure for enabling ePWM clocks is as follows:

1. Enable ePWM module clocks using the Pinmux control registers for each ePWM module instance
2. Set TBCLKSYNC = 0. This will stop the time-base clock within any enabled ePWM module.
3. Configure ePWM modules: prescaler values and ePWM modes.
4. Set TBCLKSYNC = 1.

A time-base synchronization scheme connects all of the ePWM modules on a device. Each ePWM module has a synchronization input (EPWMxSYNCl) and a synchronization output (EPWMxSYNCO). The input synchronization for the first instance (ePWM1) comes from an external pin. The synchronization connections for the remaining ePWM modules are shown in [Figure 18-6](#).



### Figure 18-6. Time-Base Counter Synchronization Scheme



Each ePWM module can be configured to use or ignore the synchronization input. If the TBCTL[PHSEN] bit is set, then the time-base counter (TBCTR) of the ePWM module will be automatically loaded with the phase register (TBPHS) contents when one of the following conditions occur:

- **EPWMxSYNCl: Synchronization Input Pulse:**

The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (TBPHS → TBCTR). This operation occurs on the next valid time-base clock (TBCLK) edge.

The delay from internal controller module to target modules is given by:

- if ( TBCLK = VCLK4):  $2 \times VCLK4$
- if ( TBCLK != VCLK4): 1 TBCLK

- **Software Forced Synchronization Pulse:**

Writing a 1 to the TBCTL[SWFSYNC] control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWMxSYNCl.

- **Digital Compare Event Synchronization Pulse:**

DCAEVT1 and DCBEVT1 digital compare events can be configured to generate synchronization pulses which have the same affect as EPWMxSYNCl.

This feature enables the ePWM module to be automatically synchronized to the time base of another ePWM module. Lead or lag phase control can be added to the waveforms generated by different ePWM modules to synchronize them. In up-down-count mode, the TBCTL[PSHDIR] bit configures the direction of the time-base counter immediately after a synchronization event. The new direction is independent of the direction prior to the synchronization event. The PSHDIR bit is ignored in count-up or count-down modes. See [Figure 18-7](#) through [Figure 18-10](#) for examples.

Clearing the TBCTL[PHSEN] bit configures the ePWM to ignore the synchronization input pulse. The synchronization pulse can still be allowed to flow-through to the EPWMxSYNCO and be used to synchronize other ePWM modules. In this way, you can set up a controller time-base (for example, ePWM1) and downstream modules (ePWM2 - ePWMx) may elect to run in synchronization with the controller.

#### 18.2.2.4 Phase Locking the Time-Base Clocks of Multiple ePWM Modules

The TBCLKSYNC bit can be used to globally synchronize the time-base clocks of all enabled ePWM modules on a device. When TBCLKSYNC = 0, the time-base clock of all ePWM modules is stopped (default). When TBCLKSYNC = 1, all ePWM time-base clocks are started with the rising edge of TBCLK aligned. For perfectly synchronized TBCLKs, the prescaler bits in the TBCTL register of each ePWM module must be set identically. The proper procedure for enabling the ePWM clocks is as follows:

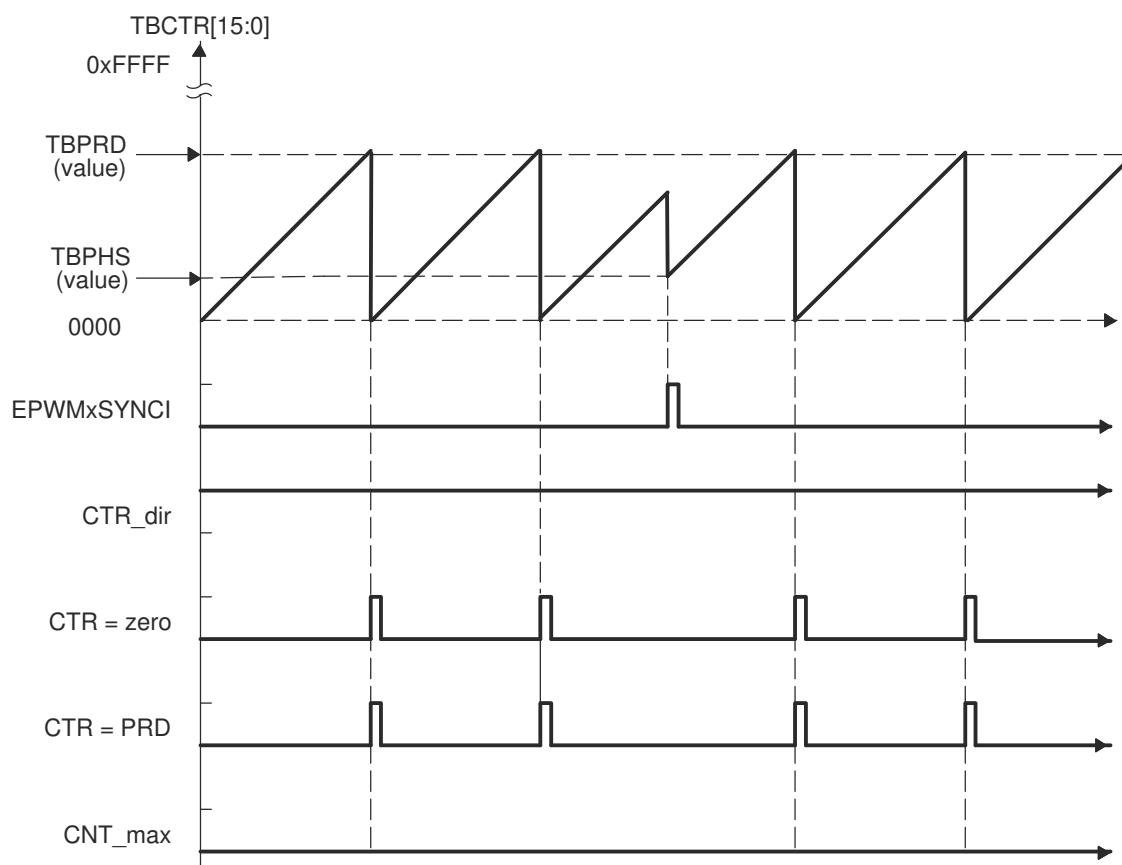
1. Enable ePWM module clocks using the IOMM control registers for each ePWM module instance
2. Set TBCLKSYNC = 0. This will stop the time-base clock within any enabled ePWM module.
3. Configure ePWM modules: prescaler values and ePWM modes.
4. Set TBCLKSYNC = 1.

### 18.2.2.5 Time-Base Counter Modes and Timing Waveforms

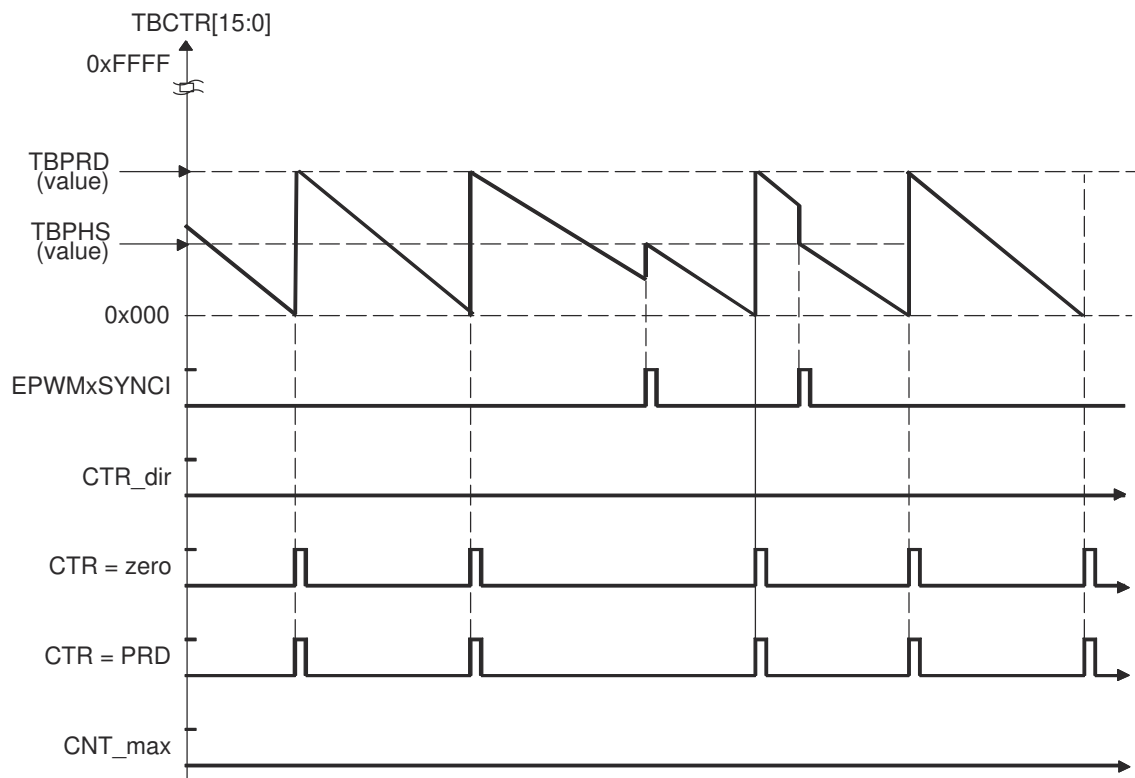
The time-base counter operates in one of four modes:

- Up-count mode which is asymmetrical.
- Down-count mode which is asymmetrical.
- Up-down-count which is symmetrical
- Frozen where the time-base counter is held constant at the current value

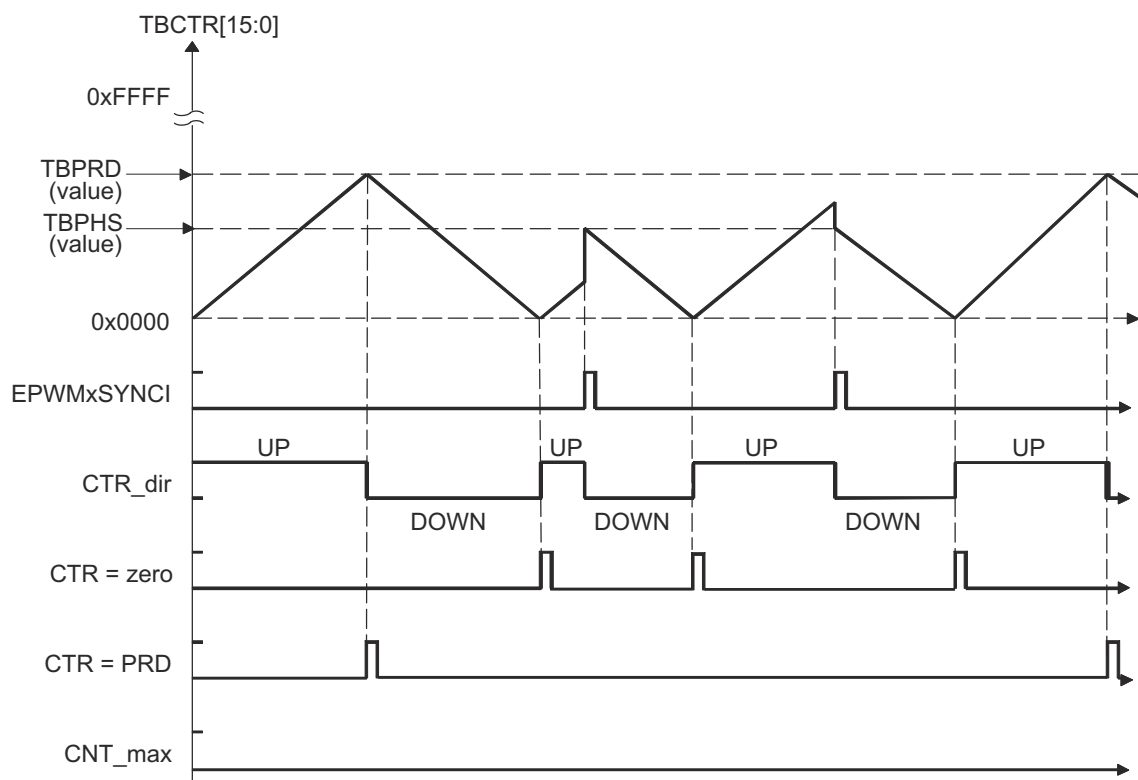
To illustrate the operation of the first three modes, the following timing diagrams show when events are generated and how the time-base responds to an EPWMxSYNCI signal.



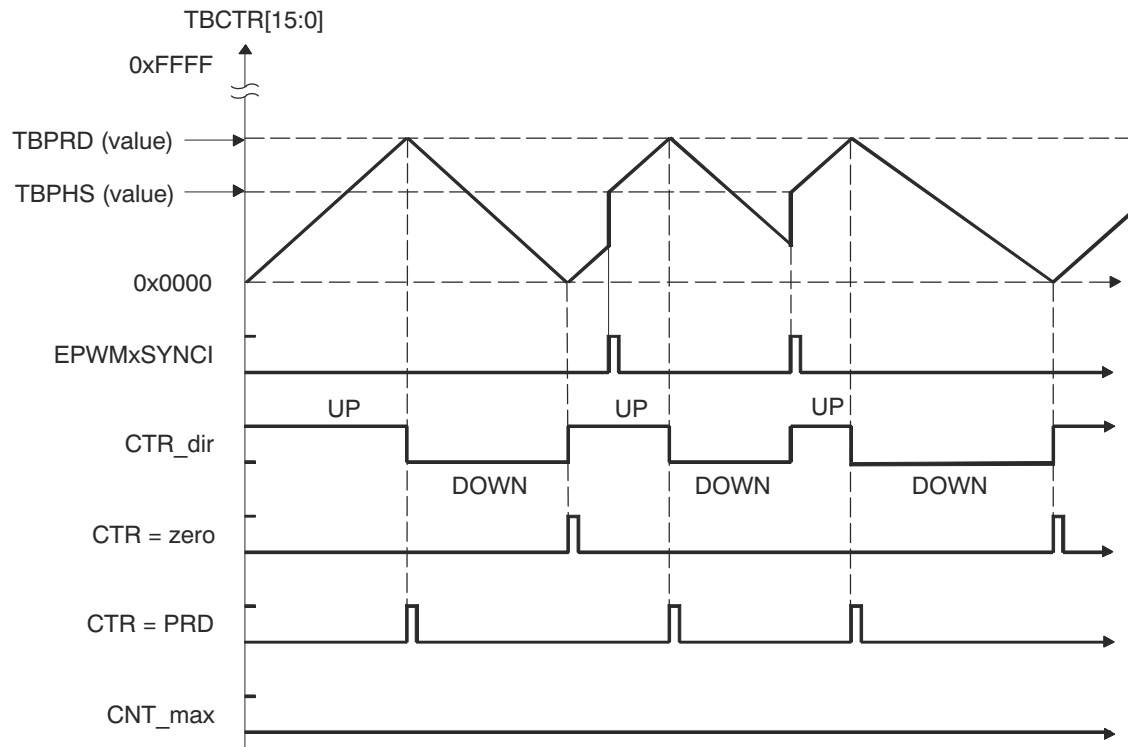
**Figure 18-7. Time-Base Up-Count Mode Waveforms**



**Figure 18-8. Time-Base Down-Count Mode Waveforms**



**Figure 18-9. Time-Base Up-Down-Count Waveforms, TBCTL[PHSDIR = 0] Count Down On Synchronization Event**



**Figure 18-10. Time-Base Up-Down Count Waveforms, TBCTL[PHSDIR = 1] Count Up On Synchronization Event**

### 18.2.3 Counter-Compare (CC) Submodule

Figure 18-11 illustrates the counter-compare submodule within the ePWM.

Figure 18-12 shows the basic structure of the counter-compare submodule.

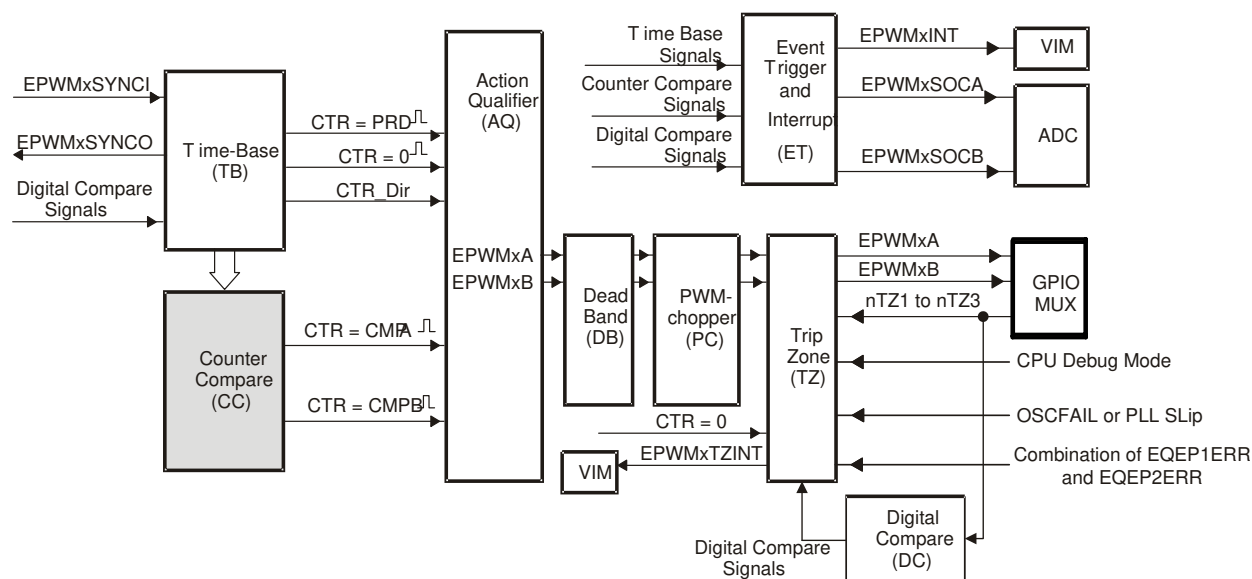


Figure 18-11. Counter-Compare Submodule

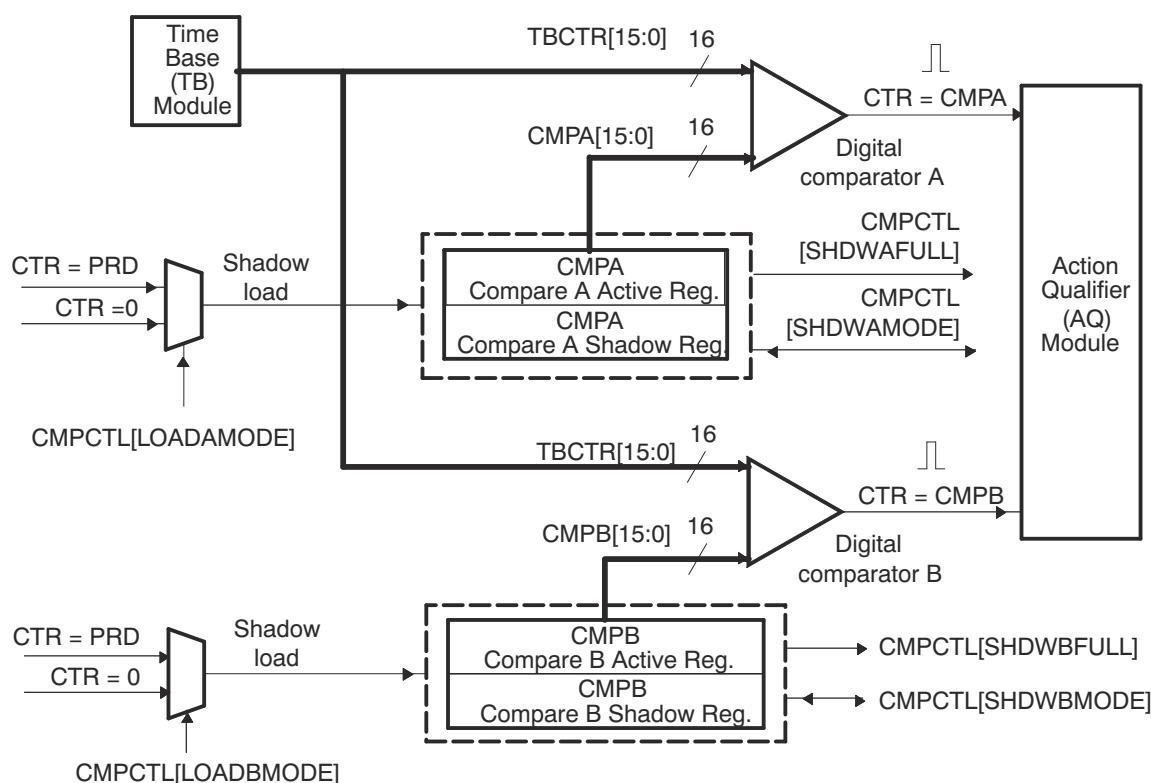


Figure 18-12. Detailed View of the Counter-Compare Submodule

### 18.2.3.1 Purpose of the Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (CMPA) and counter-compare B (CMPB) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare:

- Generates events based on programmable time stamps using the CMPA and CMPB registers
  - CTR = CMPA: Time-base counter equals counter-compare A register (TBCTR = CMPA).
  - CTR = CMPB: Time-base counter equals counter-compare B register (TBCTR = CMPB)
- Controls the PWM duty cycle if the action-qualifier submodule is configured appropriately
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

### 18.2.3.2 Controlling and Monitoring the Counter-Compare Submodule

The counter-compare submodule operation is controlled and monitored by the registers shown in [Table 18-5](#):

**Table 18-5. Counter-Compare Submodule Registers**

Register Name	Address Offset	Shadowed	Description
CMPCTL	0x000C	No	Counter-Compare Control Register.
CMPA	0x0010	Yes	Counter-Compare A Register
CMPB	0x0016	Yes	Counter-Compare B Register

The key signals associated with the counter-compare submodule are described in [Table 18-6](#).

**Table 18-6. Counter-Compare Submodule Key Signals**

Signal	Description of Event	Registers Compared
CTR = CMPA	Time-base counter equal to the active counter-compare A value	TBCTR = CMPA
CTR = CMPB	Time-base counter equal to the active counter-compare B value	TBCTR = CMPB
CTR = PRD	Time-base counter equal to the active period. Used to load active counter-compare A and B registers from the shadow register	TBCTR = TBPRD
CTR = ZERO	Time-base counter equal to zero. Used to load active counter-compare A and B registers from the shadow register	TBCTR = 0x0000

### 18.2.3.3 Operational Highlights for the Counter-Compare Submodule

The counter-compare submodule is responsible for generating two independent compare events based on two compare registers:

1. CTR = CMPA: Time-base counter equal to counter-compare A register (TBCTR = CMPA).
2. CTR = CMPB: Time-base counter equal to counter-compare B register (TBCTR = CMPB).

For up-count or down-count mode, each event occurs only once per cycle. For up-down-count mode each event occurs twice per cycle if the compare value is between 0x0000-TBPRD and once per cycle if the compare value is equal to 0x0000 or equal to TBPRD. These events are fed into the action-qualifier submodule where they are qualified by the counter direction and converted into actions if enabled. Refer to [Section 18.2.4.1](#) for more details.

The counter-compare registers CMPA and CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occur at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. Which register is written to or read from is determined by the CMPCTL[SHDWAMODE] and CMPCTL[SHDWBMODE] bits. These bits enable and disable the CMPA shadow register and CMPB shadow register respectively. The behavior of the two load modes is as described:

#### Shadow Mode:

The shadow mode for the CMPA is enabled by clearing the CMPCTL[SHDWAMODE] bit and the shadow register for CMPB is enabled by clearing the CMPCTL[SHDWBMODE] bit. Shadow mode is enabled by default for both CMPA and CMPB.

If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events as specified by the CMPCTL[LOADAMODE] and CMPCTL[LOADBMODE] register bits:

- CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD).
- CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000)
- Both CTR = PRD and CTR = Zero

Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.

#### Immediate Load Mode:

If immediate load mode is selected (that is, TBCTL[SHADWAMODE] = 1 or TBCTL[SHADWBMODE] = 1), then a read from or a write to the register will go directly to the active register.

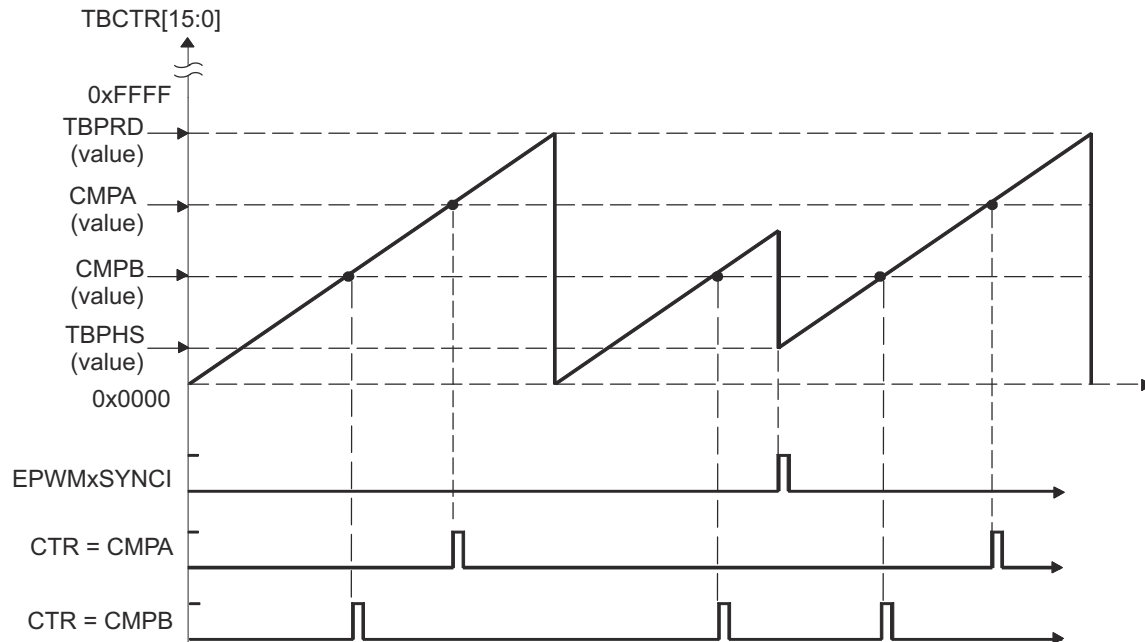


### 18.2.3.4 Count Mode Timing Waveforms

The counter-compare module can generate compare events in all three count modes:

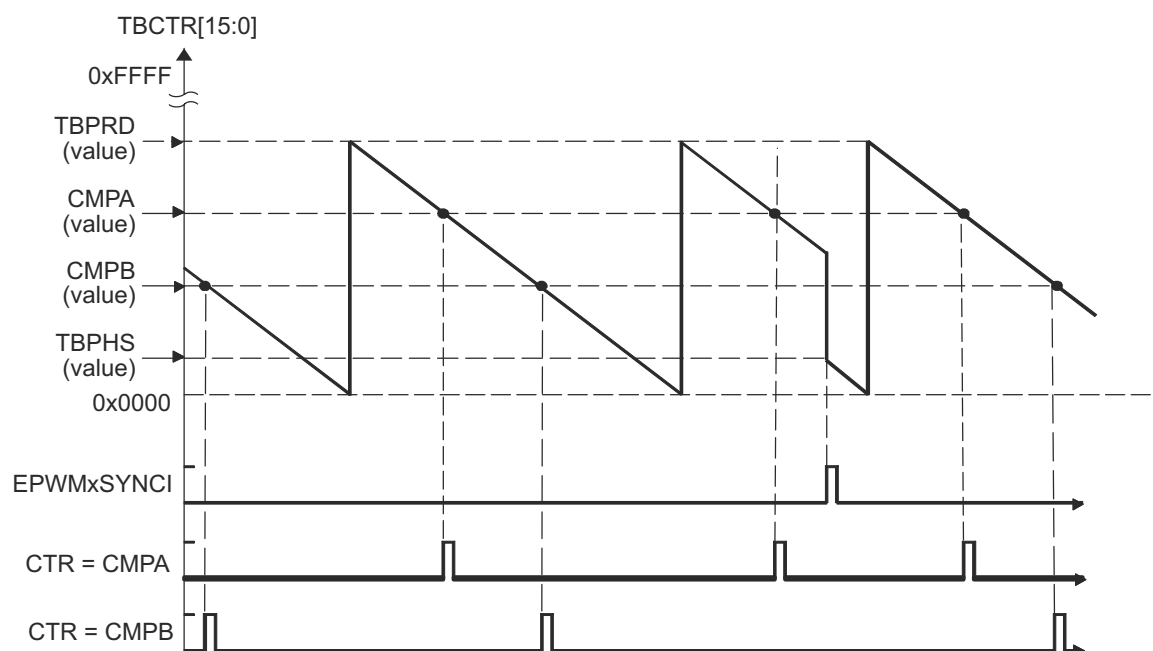
- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.
- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the first three modes, the timing diagrams in [Figure 18-13](#) through [Figure 18-16](#) show when events are generated and how the EPWMxSYNCI signal interacts.

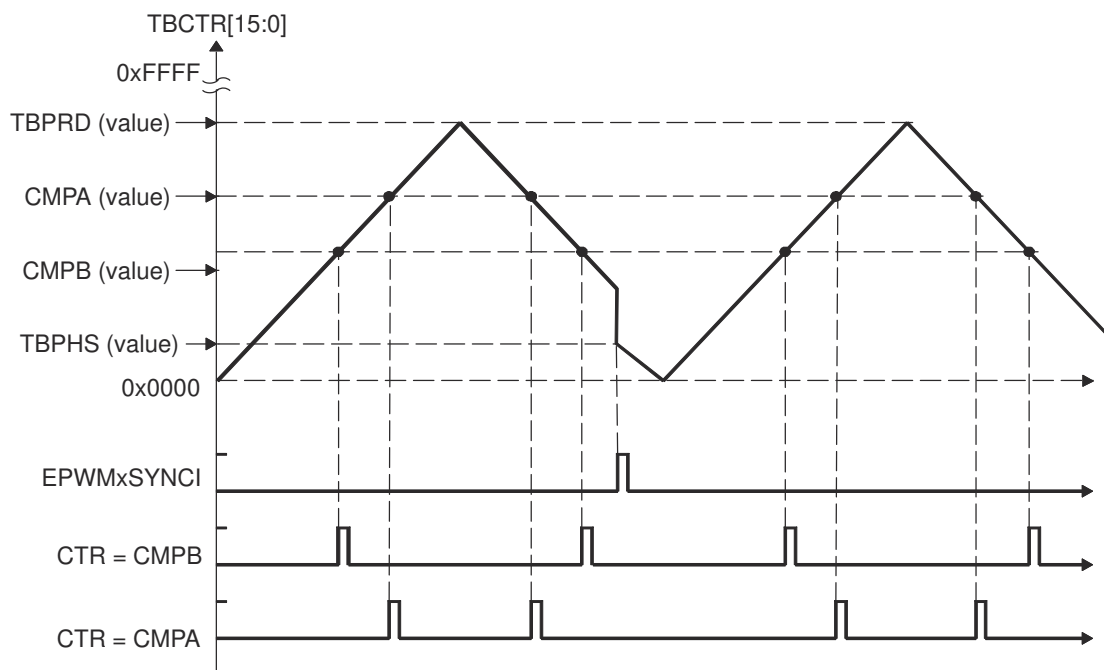


An EPWMxSYNCI external synchronization event can cause a discontinuity in the TBCTR count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

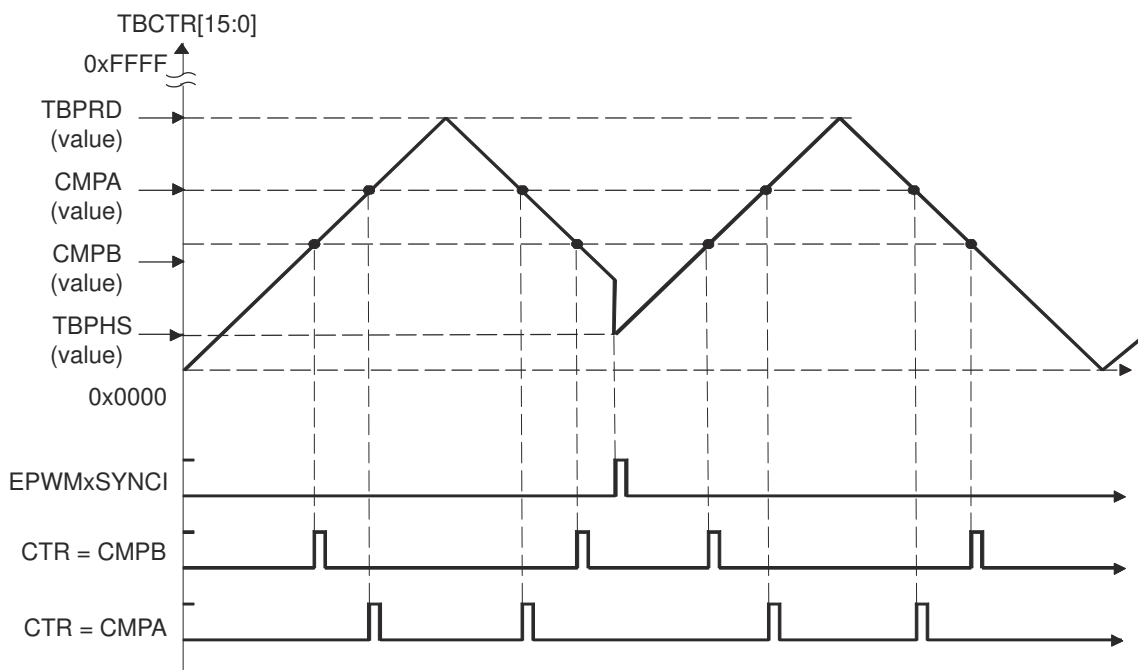
**Figure 18-13. Counter-Compare Event Waveforms in Up-Count Mode**



**Figure 18-14. Counter-Compare Events in Down-Count Mode**



**Figure 18-15. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 0] Count Down On Synchronization Event**

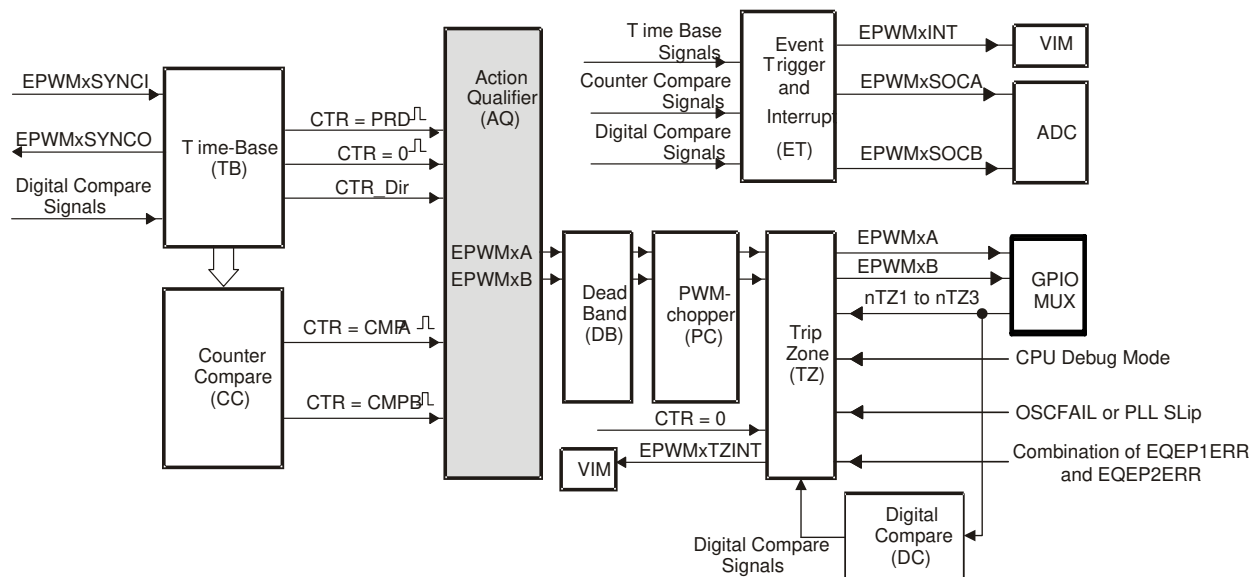


**Figure 18-16. Counter-Compare Events In Up-Down-Count Mode, TBCTL[PHSDIR = 1] Count Up On Synchronization Event**

### 18.2.4 Action-Qualifier (AQ) Submodule

Figure 18-17 shows the action-qualifier (AQ) submodule (see shaded block) in the ePWM system.

The action-qualifier submodule has the most important role in waveform construction and PWM generation. It decides which events are converted into various action types, thereby producing the required switched waveforms at



**Figure 18-17. Action-Qualifier Submodule**

#### 18.2.4.1 Purpose of the Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - CTR = PRD: Time-base counter equal to the period (TBCTR = TBPRD)
  - CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000)
  - CTR = CMPA: Time-base counter equal to the counter-compare A register (TBCTR = CMPA)
  - CTR = CMPB: Time-base counter equal to the counter-compare B register (TBCTR = CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when it is decreasing

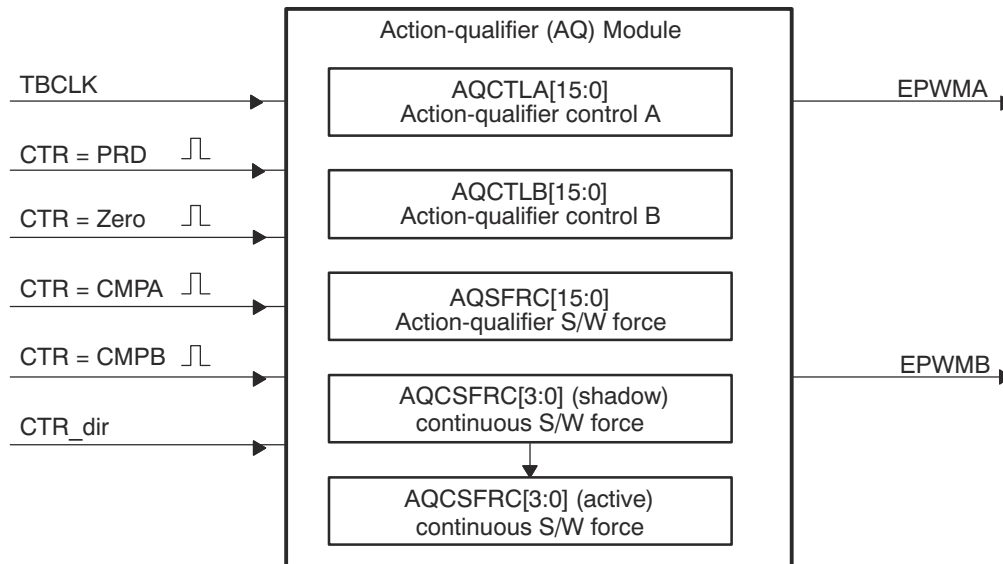
#### 18.2.4.2 Action-Qualifier Submodule Control and Status Register Definitions

The action-qualifier submodule operation is controlled and monitored via the registers in Table 18-7.

**Table 18-7. Action-Qualifier Submodule Registers**

Register Name	Address Offset	Shadowed	Description
AQCTLA	0x0014	No	Action-Qualifier Control Register For Output A (EPWMxA)
AQCTLB	0x001A	No	Action-Qualifier Control Register For Output B (EPWMxB)
AQSFR	0x0018	No	Action-Qualifier Software Force Register
AQCSFR	0x001E	Yes	Action-Qualifier Continuous Software Force

The action-qualifier submodule is based on event-driven logic. It can be thought of as a programmable cross switch with events at the input and actions at the output, all of which are software controlled via the set of registers shown in Table 18-7.



**Figure 18-18. Action-Qualifier Submodule Inputs and Outputs**

For convenience, the possible input events are summarized again in [Table 18-8](#).

**Table 18-8. Action-Qualifier Submodule Possible Input Events**

Signal	Description	Registers Compared
CTR = PRD	Time-base counter equal to the period value	TBCTR = TBPRD
CTR = Zero	Time-base counter equal to zero	TBCTR = 0x0000
CTR = CMPA	Time-base counter equal to the counter-compare A	TBCTR = CMPA
CTR = CMPB	Time-base counter equal to the counter-compare B	TBCTR = CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by registers AQSFR and AQCSFR.

The action-qualifier submodule controls how the two outputs EPWMxA and EPWMxB behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs EPWMxA and EPWMxB are:

- **Set High:**

Set output EPWMxA or EPWMxB to a high level.

- **Clear Low:**

Set output EPWMxA or EPWMxB to a low level.

- **Toggle:**





















If EPWMxA or EPWMxB is currently pulled high, then pull the output low. If EPWMxA or EPWMxB is currently pulled low, then pull the output high.

- **Do Nothing:**

Keep outputs EPWMxA and EPWMxB at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWMxA and EPWMxB outputs, this event can still trigger interrupts and ADC start of conversion. See the Event-trigger Submodule description in [Section 18.2.8](#) for details.

Actions are specified independently for either output (EPWMxA or EPWMxB). Any or all events can be configured to generate actions on a given output. For example, both CTR = CMPA and CTR = CMPB can operate on output EPWMxA. All qualifier actions are configured via the control registers found at the end of this section.

For clarity, the drawings in this document use a set of symbolic actions. These symbols are summarized in [Figure 18-19](#). Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and their time positions are programmed via the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"; it is the default at reset.

S/W force	TB Counter equals:				Actions
	Zero	Comp A	Comp B	Period	
					Do Nothing
					Clear Low
					Set High
					Toggle

**Figure 18-19. Possible Action-Qualifier Actions for EPWMxA and EPWMxB Outputs**

### 18.2.4.3 Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down-count mode are shown in [Table 18-9](#). A priority level of 1 is the highest priority and level 7 is the lowest. The priority changes slightly depending on the direction of TBCTR.

**Table 18-9. Action-Qualifier Event Priority for Up-Down-Count Mode**

Priority Level	Event If TBCTR is Incrementing TBCTR = Zero up to TBCTR = TBPRD	Event If TBCTR is Decrementing TBCTR = TBPRD down to TBCTR = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4	Counter equals zero	Counter equals period (TBPRD)
5	Counter equals CMPB on down-count (CBD)	Counter equals CMPB on up-count (CBU)
6 (Lowest)	Counter equals CMPA on down-count (CAD)	Counter equals CMPA on up-count (CBU)

[Table 18-10](#) shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up and thus down-count events will never be taken.

**Table 18-10. Action-Qualifier Event Priority for Up-Count Mode**

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

[Table 18-11](#) shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down and thus up-count events will never be taken.

**Table 18-11. Action-Qualifier Event Priority for Down-Count Mode**

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	Counter equal to CMPB on down-count (CBD)
4	Counter equal to CMPA on down-count (CAD)
5 (Lowest)	Counter equal to period (TBPRD)

It is possible to set the compare value greater than the period. In this case the action will take place as shown in [Table 18-12](#).

**Table 18-12. Behavior if CMPA/CMPB is Greater than the Period**

Counter Mode	Compare on Up-Count Event CAD/CBD	Compare on Down-Count Event CAD/CBD
Up-Count Mode	<p>If <math>CMPA/CMPB \leq TBPRD</math> period, then the event occurs on a compare match (<math>TBCTR = CMPA</math> or <math>CMPB</math>).</p> <p>If <math>CMPA/CMPB &gt; TBPRD</math>, then the event will not occur.</p>	Never occurs.
Down-Count Mode	Never occurs.	<p>If <math>CMPA/CMPB &lt; TBPRD</math>, the event will occur on a compare match (<math>TBCTR = CMPA</math> or <math>CMPB</math>).</p> <p>If <math>CMPA/CMPB \geq TBPRD</math>, the event will occur on a period match (<math>TBCTR = TBPRD</math>).</p>
Up-Down-Count Mode	<p>If <math>CMPA/CMPB &lt; TBPRD</math> and the counter is incrementing, the event occurs on a compare match (<math>TBCTR = CMPA</math> or <math>CMPB</math>).</p> <p>If <math>CMPA/CMPB \geq TBPRD</math>, the event will occur on a period match (<math>TBCTR = TBPRD</math>).</p>	<p>If <math>CMPA/CMPB &lt; TBPRD</math> and the counter is decrementing, the event occurs on a compare match (<math>TBCTR = CMPA</math> or <math>CMPB</math>).</p> <p>If <math>CMPA/CMPB \geq TBPRD</math>, the event occurs on a period match (<math>TBCTR = TBPRD</math>).</p>

#### 18.2.4.4 Waveforms for Common Configurations

##### Note

The waveforms in this document show the ePWMs behavior for a static compare register value. In a running system, the active compare registers (CMPA and CMPB) are typically updated from their respective shadow registers once every period. The user specifies when the update will take place; either when the time-base counter reaches zero or when the time-base counter reaches period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

##### Use up-down-count mode to generate a symmetric PWM:

- If you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1.
- If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to  $TBPRD - 1$ .

This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

##### Use up-down-count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load CMPA/CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

##### When using up-count mode to generate an asymmetric PWM:

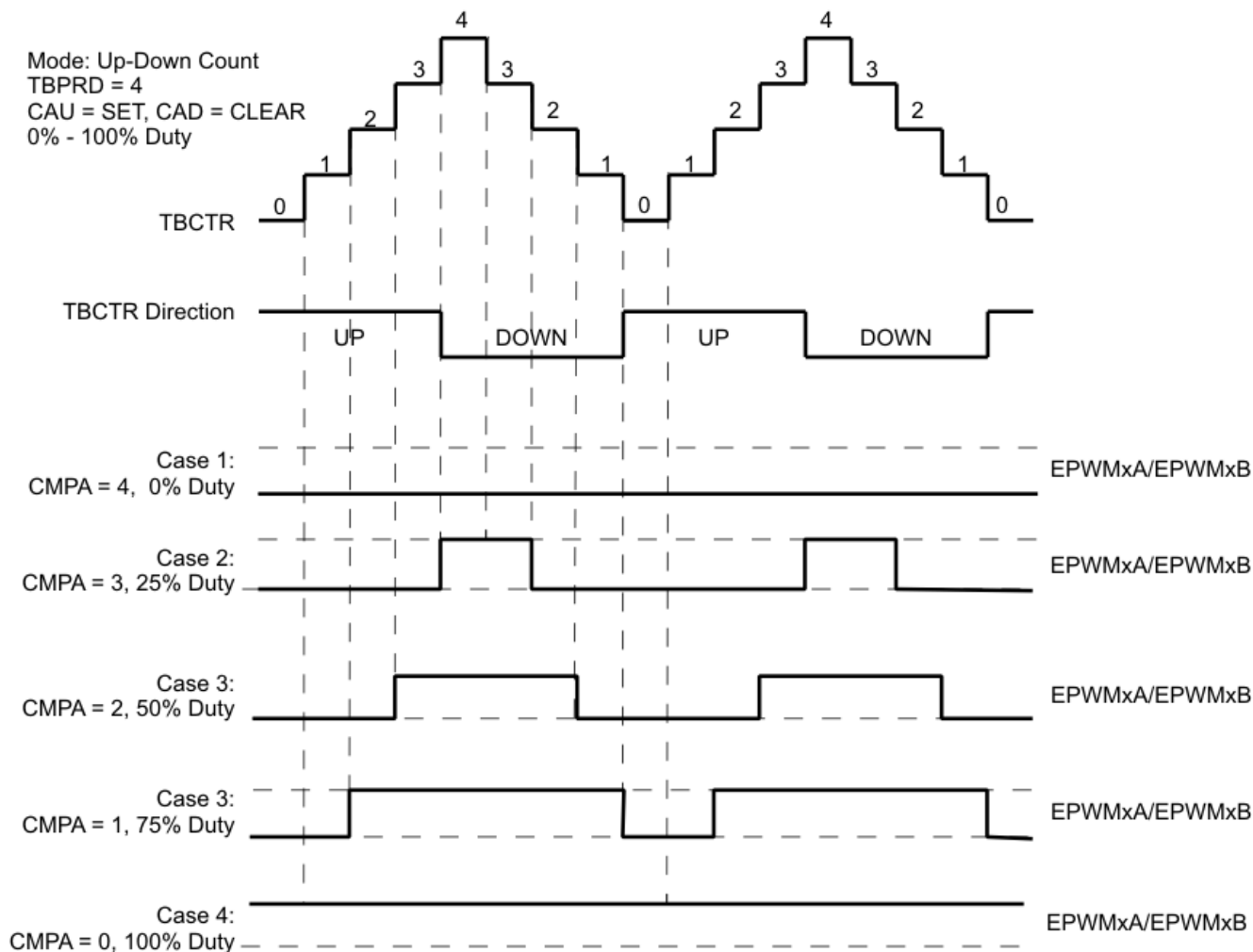
- To achieve 0-100% asymmetric PWM use the following configuration: Load CMPA/CMPB on TBPRD. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to  $TBPRD + 1$  to achieve 0-100% PWM duty.

See the *Using Enhanced Pulse Width Modulator (ePWM) Module for 0-100% Duty Cycle Control* Application Report ([SPRAA11](#))



Figure 18-20 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCTR. In this mode 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing the CMPA match will pull the PWM output high. Likewise, when the counter is decrementing the compare match will pull the PWM signal low. When CMPA = 0, the PWM signal is low for the entire period giving the 0% duty waveform. When CMPA = TBPRD, the PWM signal is high achieving 100% duty.

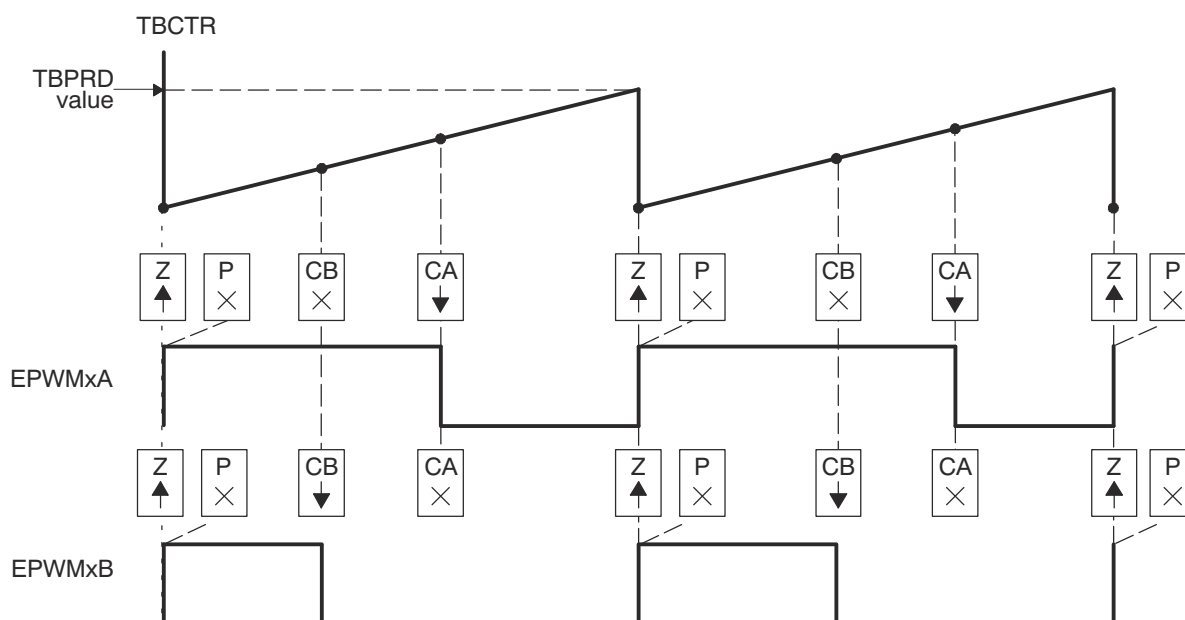
When using this configuration in practice, if you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD - 1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.



**Figure 18-20. Up-Down-Count Mode Symmetrical Waveform**

The PWM waveforms in [Figure 18-21](#) through [Figure 18-26](#) show some common action-qualifier configurations. The C-code samples in [Example 18-1](#) through [Example 18-6](#) shows how to configure an ePWM module for each case. Some conventions used in the figures and examples are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in their respective registers. The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB
- EPWMxA and EPWMxB refer to the output signals from ePWMx
- Up-Down means Count-up-and-down mode, Up means up-count mode and Dwn means down-count mode
- Sym = Symmetric, Asym = Asymmetric



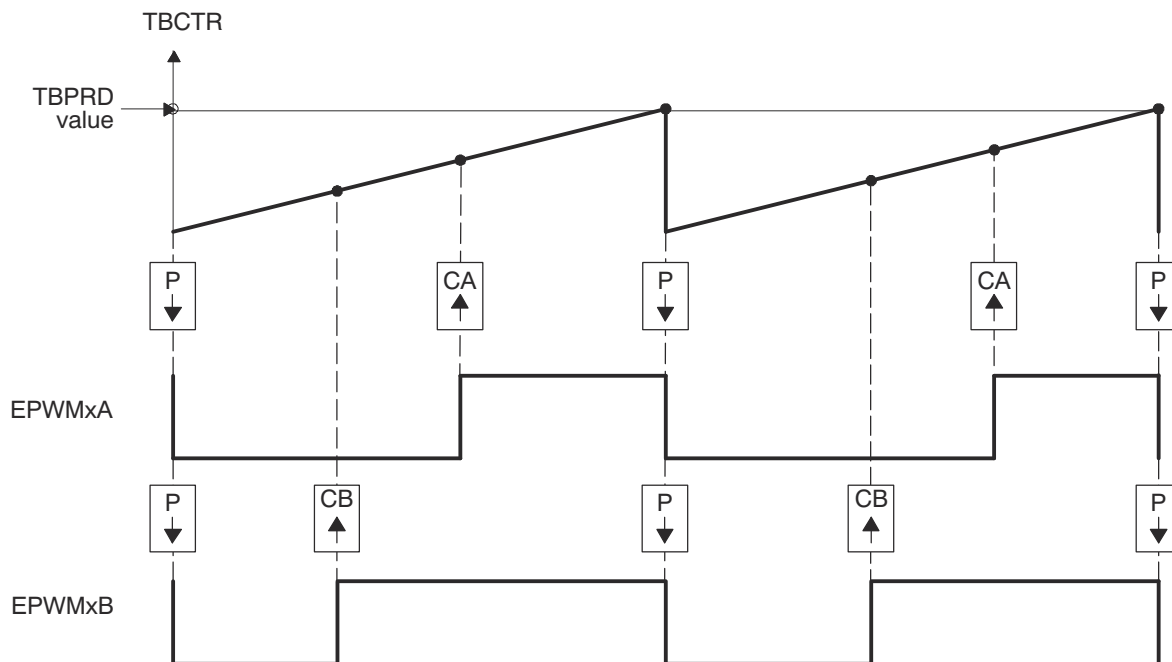
- PWM period =  $(TBPRD + 1) \times T_{TBCLK}$
- Duty modulation for EPWMxA is set by CMPA, and is active high (that is, high time duty proportional to CMPA).
- Duty modulation for EPWMxB is set by CMPB and is active high (that is, high time duty proportional to CMPB).
- The "Do Nothing" actions ( X ) are shown for completeness, but will not be shown on subsequent diagrams.
- Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

**Figure 18-21. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB—Active High**

[Example 18-1](#) contains a code sample showing initialization and run time for the waveforms in [Figure 18-21](#).

**Example 18-1. Code Sample for Figure 18-21**

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200; // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = SYSCLK
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_SET;
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B; // adjust duty for output EPWM1B
```

**18.2.4.5**


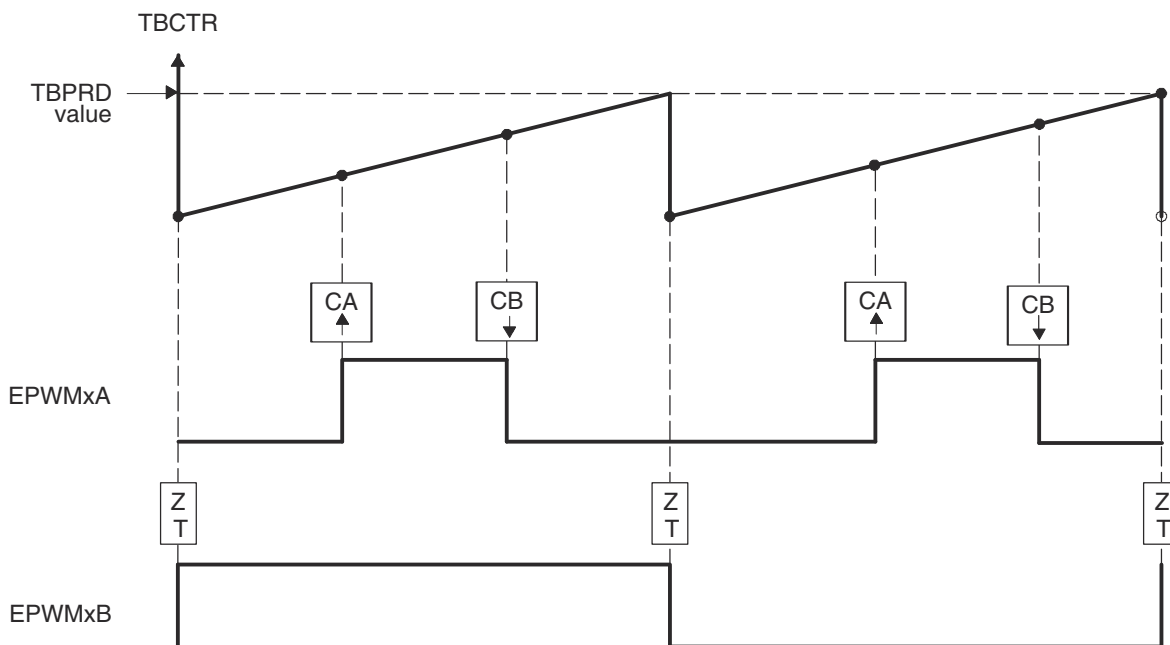
- PWM period =  $(TBPRD + 1) \times T_{TBCLK}$
- Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).
- Duty modulation for EPWMB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).
- Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCTR wraps from period to 0000.

**Figure 18-22. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWMxA and EPWMB—Active Low**

[Example 18-2](#) contains a code sample showing initialization and run time for the waveforms in [Figure 18-22](#).

**Example 18-2. Code Sample for Figure 18-22**

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350; // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 200; // Compare B = 200 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = VCLK4
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.PR = AQ_CLEAR;
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLB.bit.PR = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B; // adjust duty for output EPWM1B
```

**18.2.4.6**


- PWM frequency =  $1 / ((TBPRD + 1) \times T_{TBCLK})$
- Pulse can be placed anywhere within the PWM cycle (0000 - TBPRD)
- High time duty proportional to (CMPB - CMPA)
- EPWMxB can be used to generate a 50% duty square wave with frequency =  $1/2 \times ((TBPRD + 1) \times TBCLK)$

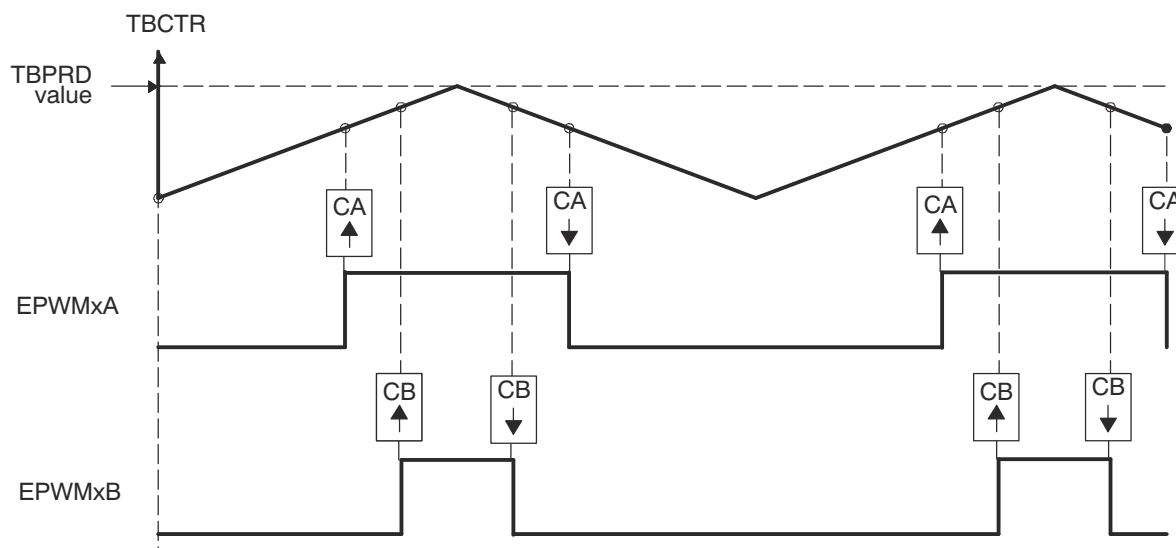
**Figure 18-23. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWMxA**

Example 18-3 contains a code sample showing initialization and run time for the waveforms in Figure 18-23.

### Example 18-3. Code Sample for Figure 18-23

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600;                // Period = 601 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 200;      // Compare A = 200 TBCLK counts
EPwm1Regs.CMPB = 400;                // Compare B = 400 TBCLK counts
EPwm1Regs.TBPHS = 0;                 // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                  // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = VCLK4
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on TBCTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CBU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_TOGGLE;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = EdgePosA; // adjust duty for output EPWM1A only
EPwm1Regs.CMPB = EdgePosB;
```

### 18.2.4.7



- PWM period =  $2 \times \text{TBPRD} \times T_{\text{TBCLK}}$
- Duty modulation for EPWMxA is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).
- Duty modulation for EPWMxB is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).
- Outputs EPWMxA and EPWMxB can drive independent power switches

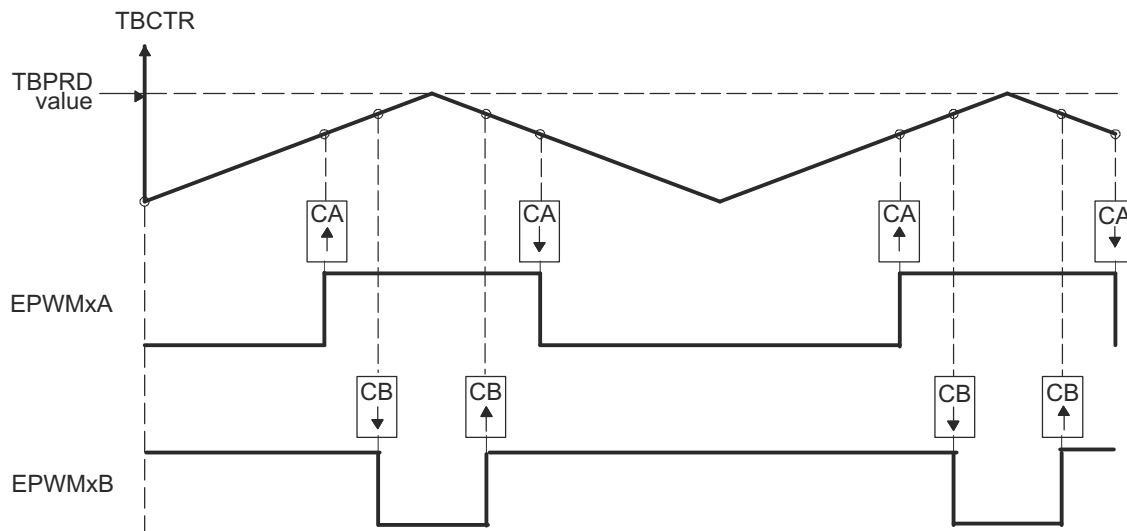
**Figure 18-24. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Active Low**

Example 18-4 contains a code sample showing initialization and run time for the waveforms in Figure 18-24.

#### Example 18-4. Code Sample for Figure 18-24

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600;           // Period = 2*600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 400; // Compare A = 400 TBCLK counts
EPwm1Regs.CMPB = 500;           // Compare B = 500 TBCLK counts
EPwm1Regs.TBPHS = 0;            // Set Phase register to zero
EPwm1Regs.TBCTR = 0;            // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetric
xEPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
xEPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = VCLK4
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET;
EPwm1Regs.AQCTLB.bit.CBD = AQ_CLEAR;
//
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B;           // adjust duty for output EPWM1B
```

#### 18.2.4.8



- PWM period =  $2 \times \text{TBPRD} \times T_{\text{TBCLK}}$
- Duty modulation for EPWMxA is set by CMPA, and is active low, that is, low time duty proportional to CMPA
- Duty modulation for EPWMxB is set by CMPB and is active high, that is, high time duty proportional to CMPB
- Outputs EPWMx can drive upper/lower (complementary) power switches
- Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

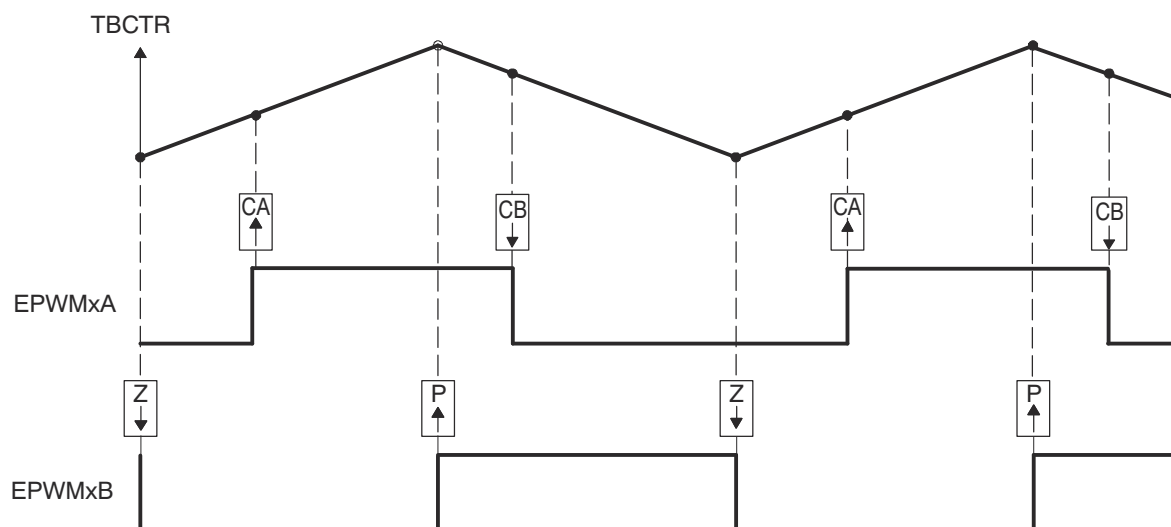
**Figure 18-25. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWMxA and EPWMxB — Complementary**

Example 18-5 contains a code sample showing initialization and run time for the waveforms in Figure 18-25.

### Example 18-5. Code Sample for Figure 18-25

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600;                // Period = 2*600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 350;      // Compare A = 350 TBCLK counts
EPwm1Regs.CMPB = 400;                // Compare B = 400 TBCLK counts
EPwm1Regs.TBPHS = 0;                 // Set Phase register to zero
EPwm1Regs.TBCTR = 0;                 // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetric
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = VCLK4
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBD = AQ_SET;
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = Duty1A;    // adjust duty for output EPWM1A
EPwm1Regs.CMPB = Duty1B;              // adjust duty for output EPWM1B
```

### 18.2.4.9



- PWM period =  $2 \times \text{TBPRD} \times \text{TBCLK}$
- Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.
- Duty modulation for EPWMxA is set by CMPA and CMPB.
- Low time duty for EPWMxA is proportional to  $(\text{CMPA} + \text{CMPB})$ .
- To change this example to active high, CMPA and CMPB actions need to be inverted (that is, Set ! Clear and Clear Set).
- Duty modulation for EPWMxB is fixed at 50% (utilizes spare action resources for EPWMxB)

**Figure 18-26. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWMxA—Active Low**



Example 18-6 contains a code sample showing initialization and run time for the waveforms in Figure 18-26.

#### Example 18-6. Code Sample for Figure 18-26

```
// Initialization Time
// =====
EPwm1Regs.TBPRD = 600; // Period = 2 * 600 TBCLK counts
EPwm1Regs.CMPA.half.CMPA = 250; // Compare A = 250 TBCLK counts
EPwm1Regs.CMPB = 450; // Compare B = 450 TBCLK counts
EPwm1Regs.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTR = 0; // clear TB counter
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetric
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.TBCTL.bit.HSPCLKDIV = TB_DIV1; // TBCLK = VCLK4
EPwm1Regs.TBCTL.bit.CLKDIV = TB_DIV1;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR = Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
EPwm1Regs.AQCTLA.bit.CBD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.PR = AQ_SET;
// Run Time
// =====
EPwm1Regs.CMPA.half.CMPA = EdgePosA; // adjust duty for output EPWM1A only
EPwm1Regs.CMPB = EdgePosB;
```

### 18.2.5 Dead-Band Generator (DB) Submodule

Figure 18-27 illustrates the dead-band submodule within the ePWM module.

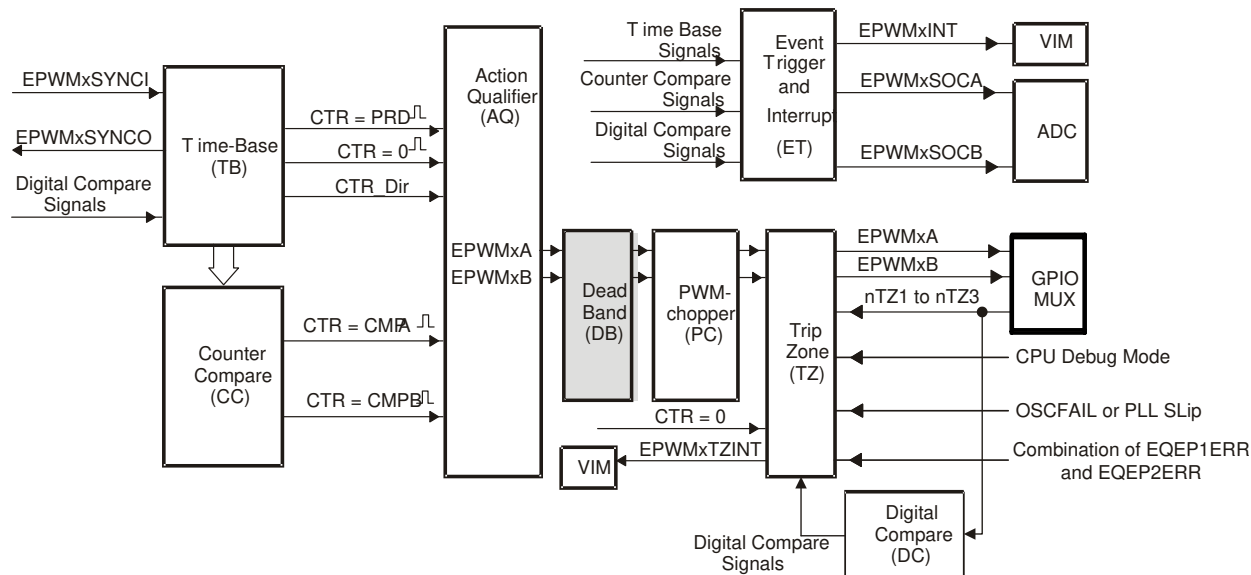


Figure 18-27. Dead\_Band Submodule

#### 18.2.5.1 Purpose of the Dead-Band Submodule

Section 18.2.4 discussed how it is possible to generate the required dead-band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical

edge delay-based dead-band with polarity control is required, then the dead-band submodule described here should be used.

The key functions of the dead-band module are:

- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

### 18.2.5.2 Controlling and Monitoring the Dead-Band Submodule

The dead-band submodule operation is controlled and monitored via the following registers:

**Table 18-13. Dead-Band Generator Submodule Registers**

Register Name	Address Offset	Shadowed	Description
DBCTL	0x001C	No	Dead-Band Control Register
DBRED	0x0022	No	Dead-Band Rising Edge Delay Count Register
DBFED	0x0020	No	Dead-Band Falling Edge Delay Count Register

### 18.2.5.3 Operational Highlights for the Dead-Band Submodule

The following sections provide the operational highlights.

The dead-band submodule has two groups of independent selection options as shown in [Figure 18-28](#).

#### • Input Source Selection:

The input signals to the dead-band module are the output signals from the action-qualifier. In this section they will be referred to as EPWMxA In and EPWMxB In. Using the DBCTL[IN\_MODE] control bits, the signal source for each delay, falling-edge or rising-edge, can be selected:

- In is the source for both falling-edge and rising-edge delay. This is the default mode.
- In is the source for falling-edge delay, EPWMxB In is the source for rising-edge delay.
- In is the source for rising edge delay, In is the source for falling-edge delay.
- In is the source for both falling-edge and rising-edge delay.

#### • Half Cycle Clocking:

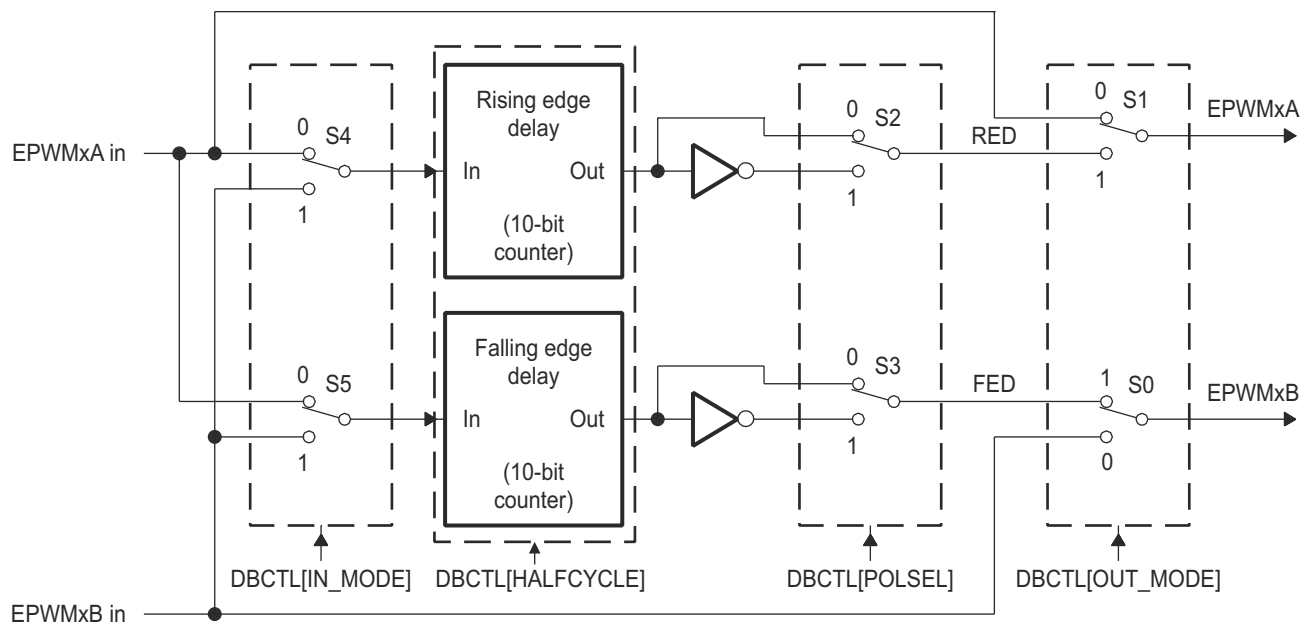
The dead-band submodule can be clocked using half cycle clocking to double the resolution (that is, counter clocked at 2× TBCLK)

#### • Output Mode Control:

The output mode is configured by way of the DBCTL[OUT\_MODE] bits. These bits determine if the falling-edge delay, rising-edge delay, neither, or both are applied to the input signals.

#### • Polarity Control:

The polarity control (DBCTL[POLSEL]) allows you to specify whether the rising-edge delayed signal and/or the falling-edge delayed signal is to be inverted before being sent out of the dead-band submodule.



**Figure 18-28. Configuration Options for the Dead-Band Submodule**

Although all combinations are supported, not all are typical usage modes. [Table 18-14](#) documents some classical dead-band configurations. These modes assume that the DBCTL[IN\_MODE] is configured such that In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in [Table 18-14](#) fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED)**

Allows you to fully disable the dead-band submodule from the PWM signal path.

- **Mode 2-5: Classical Dead-Band Polarity Settings:**

These represent typical polarity configurations that should address all the active high/low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in [Figure 18-29](#). Note that to generate equivalent waveforms to [Figure 18-29](#), configure the action-qualifier submodule to generate the signal as shown for .

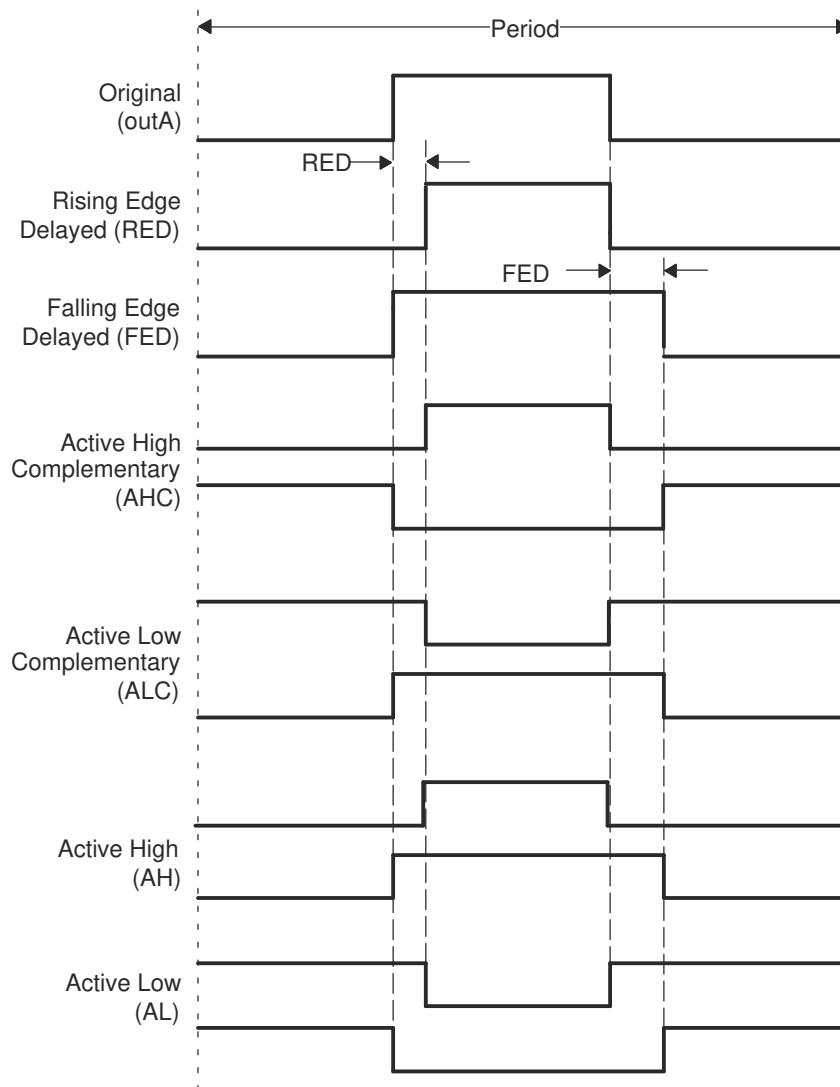
- **Mode 6: Bypass rising-edge-delay and Mode 7: Bypass falling-edge-delay**

Finally the last two entries in [Table 18-14](#) show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

**Table 18-14. Classical Dead-Band Operating Modes**

Mode	Mode Description	DBCTL[POLSEL]		DBCTL[OUT_MODE]	
		S3	S2	S1	S0
1	EPWMxA and EPWMxB Passed Through (No Delay)	X	X	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWMxA Out = EPWMxA In (No Delay) EPWMxB Out = EPWMxA In with Falling Edge Delay	0 or 1	0 or 1	0	1
7	EPWMxA Out = EPWMxA In with Rising Edge Delay EPWMxB Out = EPWMxB In with No Delay	0 or 1	0 or 1	1	0

[Figure 18-29](#) shows waveforms for typical cases where  $0\% < \text{duty} < 100\%$ .



**Figure 18-29. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)**

The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the DBRED and DBFED registers. These are 10-bit registers and their value represents the number of time-base clock, TBCLK, periods a signal edge is delayed by. For example, the formula to calculate falling-edge-delay and rising-edge-delay are:

$$\text{FED} = \text{DBFED} \times T_{\text{TBCLK}}$$

$$\text{RED} = \text{DBRED} \times T_{\text{TBCLK}}$$

Where  $T_{\text{TBCLK}}$  is the period of TBCLK, the prescaled version of VCLK4.

For convenience, delay values for various TBCLK options are shown in [Table 18-15](#).

**Table 18-15. Dead-Band Delay Values in  $\mu\text{S}$  as a Function of DBFED and DBRED**

Dead-Band Value		Dead-Band Delay in $\mu\text{S}$		
DBFED, DBRED	TBCLK = VCLK4/1	TBCLK = VCLK4 /2	TBCLK = VCLK4/4	
1	0.02 $\mu\text{S}$	0.03 $\mu\text{S}$	0.07 $\mu\text{S}$	
5	0.08 $\mu\text{S}$	0.17 $\mu\text{S}$	0.33 $\mu\text{S}$	
10	0.17 $\mu\text{S}$	0.33 $\mu\text{S}$	0.67 $\mu\text{S}$	
100	1.67 $\mu\text{S}$	3.33 $\mu\text{S}$	6.67 $\mu\text{S}$	
200	3.33 $\mu\text{S}$	6.67 $\mu\text{S}$	13.33 $\mu\text{S}$	
400	6.67 $\mu\text{S}$	13.33 $\mu\text{S}$	26.67 $\mu\text{S}$	
500	8.33 $\mu\text{S}$	16.67 $\mu\text{S}$	33.33 $\mu\text{S}$	
600	10.00 $\mu\text{S}$	20.00 $\mu\text{S}$	40.00 $\mu\text{S}$	
700	11.67 $\mu\text{S}$	23.33 $\mu\text{S}$	46.67 $\mu\text{S}$	
800	13.33 $\mu\text{S}$	26.67 $\mu\text{S}$	53.33 $\mu\text{S}$	
900	15.00 $\mu\text{S}$	30.00 $\mu\text{S}$	60.00 $\mu\text{S}$	
1000	16.67 $\mu\text{S}$	33.33 $\mu\text{S}$	66.67 $\mu\text{S}$	

When half-cycle clocking is enabled, the formula to calculate the falling-edge-delay and rising-edge-delay becomes:

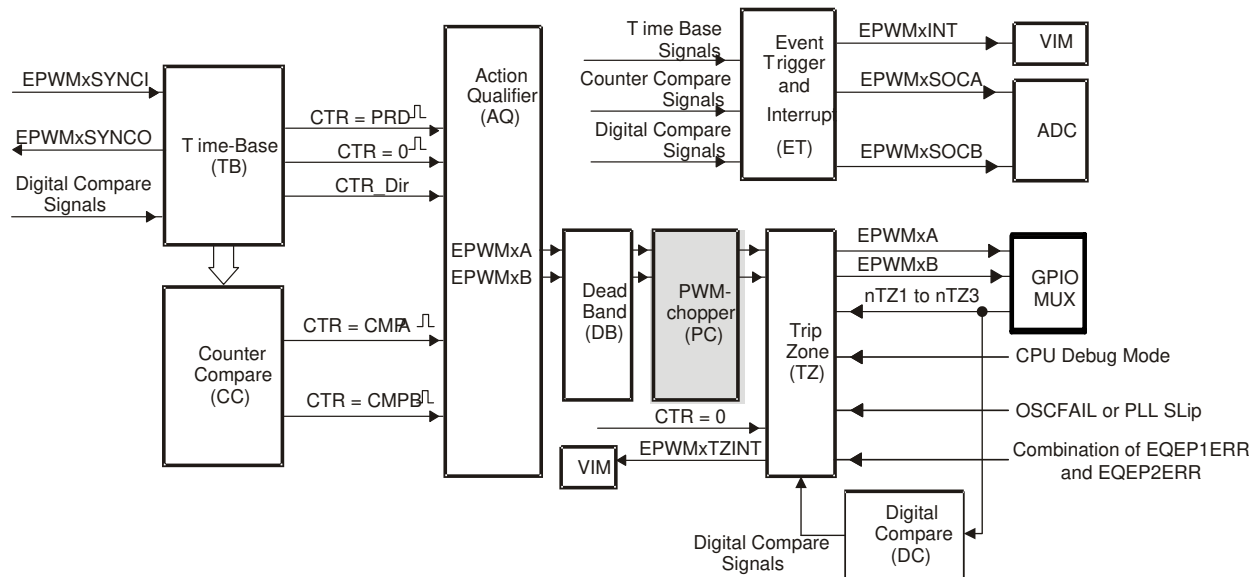
$$\text{FED} = \text{DBFED} \times T_{\text{TBCLK}}/2$$

$$\text{RED} = \text{DBRED} \times T_{\text{TBCLK}}/2$$

### 18.2.6 PWM-Chopper (PC) Submodule

Figure 18-30 illustrates the PWM-chopper (PC) submodule within the ePWM module.

The PWM-chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if you need pulse transformer-based gate drivers to control the power switching elements.



**Figure 18-30. PWM-Chopper Submodule**

#### 18.2.6.1 Purpose of the PWM-Chopper Submodule

The key functions of the PWM-chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

#### 18.2.6.2 Controlling the PWM-Chopper Submodule

The PWM-chopper submodule operation is controlled via the registers in Table 18-16.

**Table 18-16. PWM-Chopper Submodule Registers**

Register Name	Address Offset	Shadowed	Description
PCCTL	0x003E	No	PWM-chopper Control Register

#### 18.2.6.3 Operational Highlights for the PWM-Chopper Submodule

Figure 18-31 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from VCLK4. Its frequency and duty cycle are controlled via the CHPFREQ and CHPDUTY bits in the PCCTL register. The one-shot block is a feature that provides a high energy first pulse to ensure hard and fast power switch turn on, while the subsequent pulses sustain pulses, ensuring the power switch remains on. The one-shot width is programmed via the OSHTWTH bits. The PWM-chopper submodule can be fully disabled (bypassed) via the CHPEN bit.

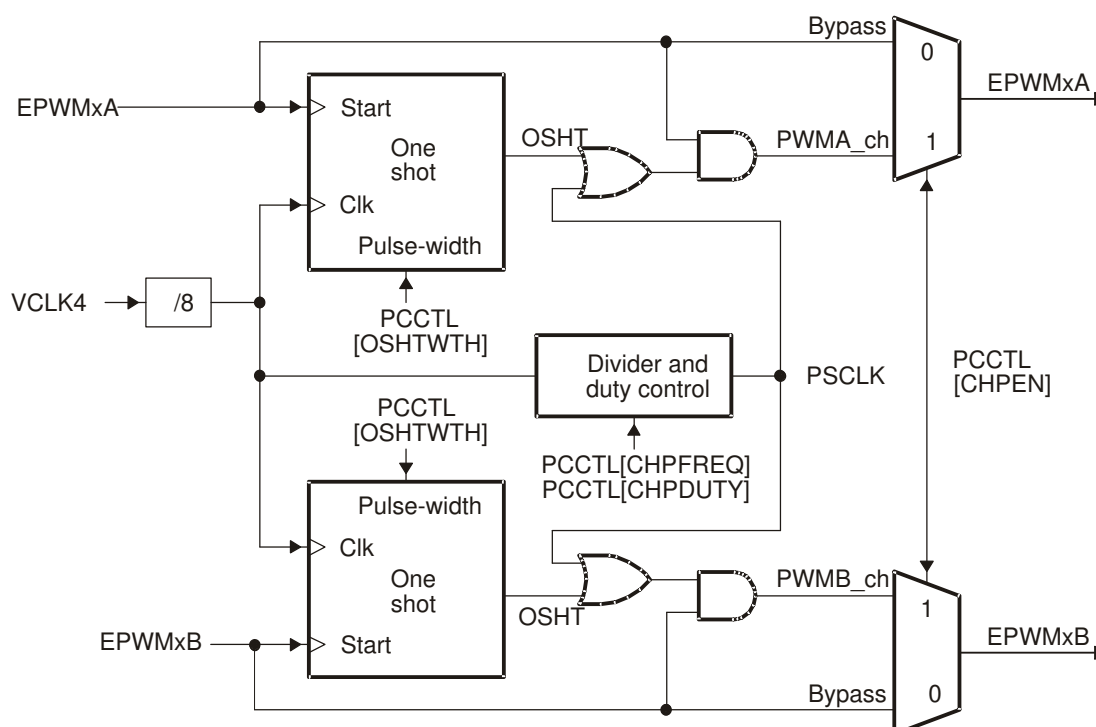


Figure 18-31. PWM-Chopper Submodule Operational Details

#### 18.2.6.4 Waveforms

Figure 18-32 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

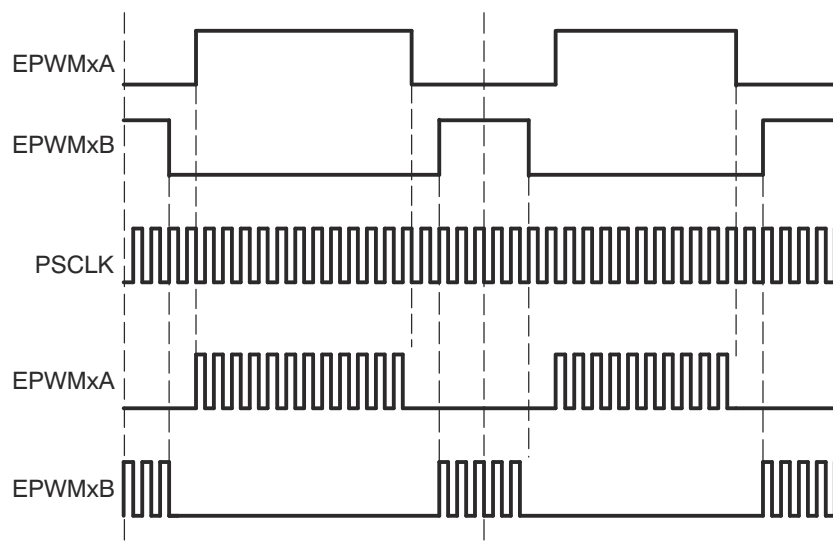


Figure 18-32. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only



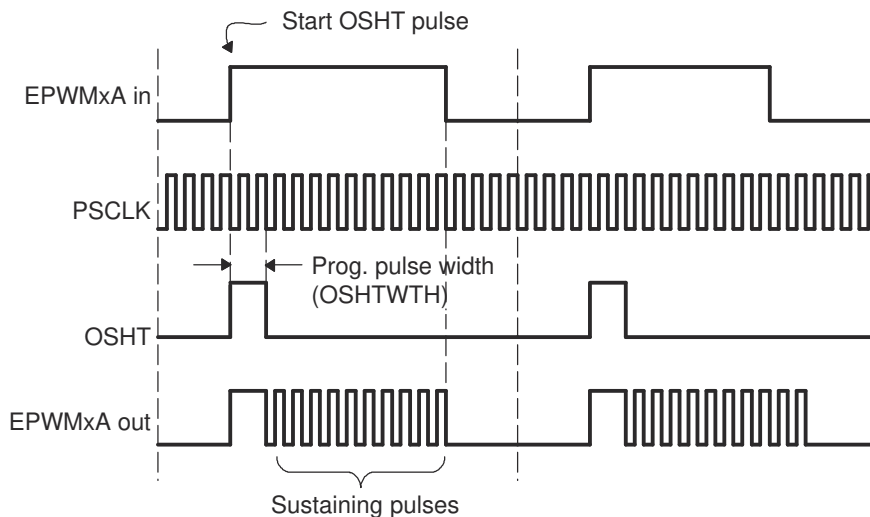
#### 18.2.6.4.1 One-Shot Pulse

The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$$T_{1stpulse} = T_{VCLK4} \times 8 \times OSHTWTH$$

Where  $T_{VCLK4}$  is the period of the system clock (VCLK4) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 18-33 shows the first and subsequent sustaining pulses and Table 18-17 gives the possible pulse width values for a VCLK4 = 100 MHz.



**Figure 18-33. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses**

**Table 18-17. Possible Pulse Width Values for  
VCLK4 = 100 MHz**

OSHTWTHz (hex)	Pulse Width (nS)
0	100
1	200
2	300
3	400
4	500
5	600
6	700
7	800
8	900
9	1000
A	1100
B	1200
C	1300
D	1400
E	1500

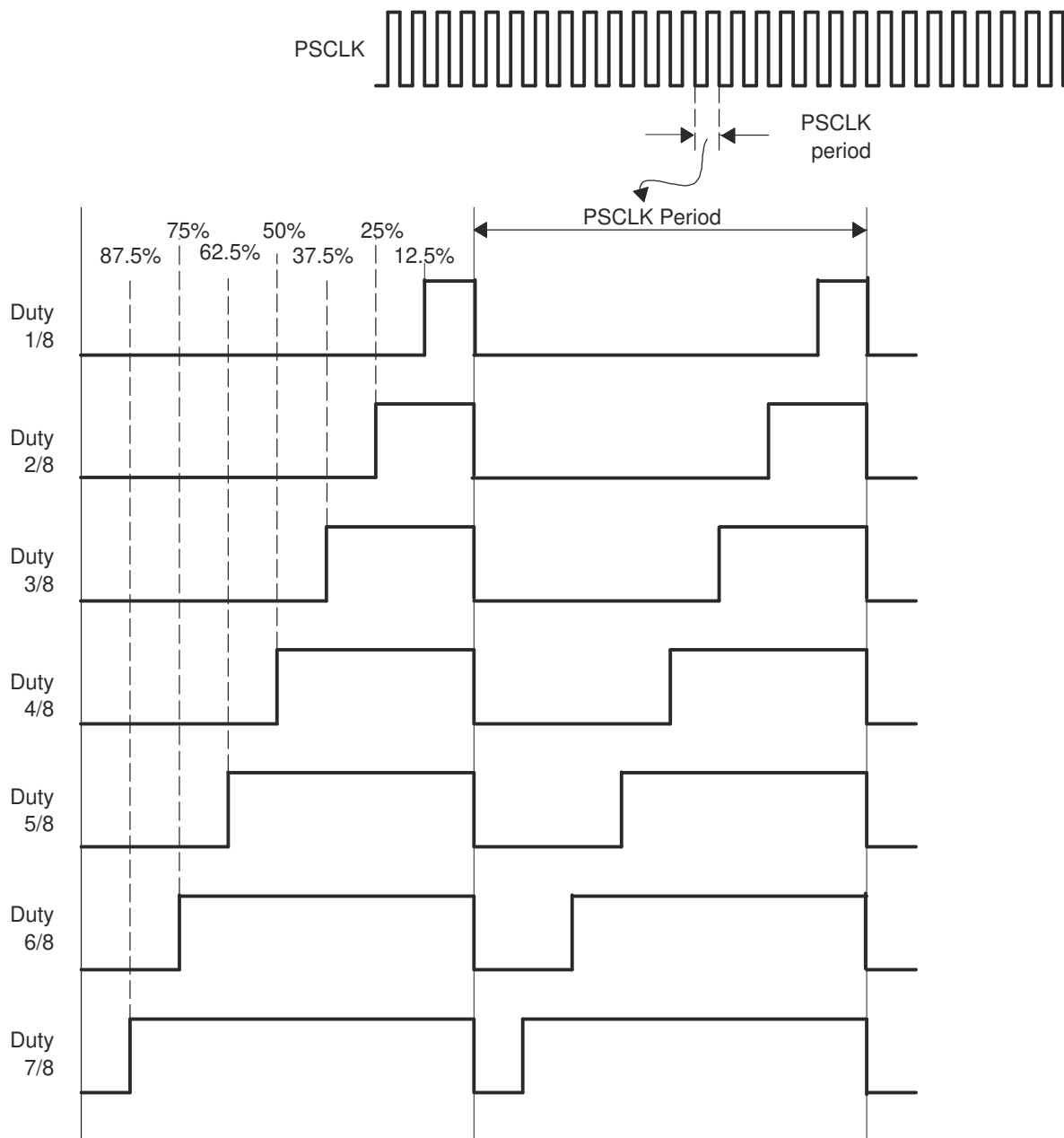
**Table 18-17. Possible Pulse Width Values for  
VCLK4 = 100 MHz (continued)**

OSHTWTHz (hex)	Pulse Width (nS)
F	1600

#### 18.2.6.4.2 Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses ensure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized via software control.

[Figure 18-34](#) shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

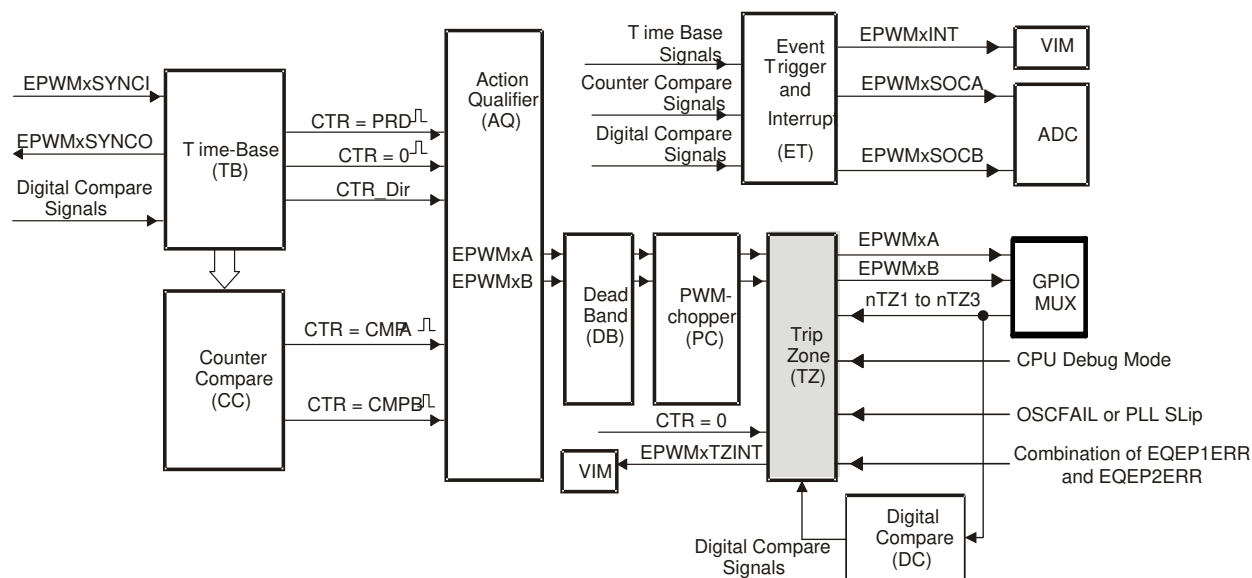


**Figure 18-34. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses**

### 18.2.7 Trip-Zone (TZ) Submodule

Figure 18-35 shows how the trip-zone (TZ) submodule fits within the ePWM module.

Each ePWM module is connected to six  $\overline{TZn}$  signals ( $\overline{TZ1}$  to  $\overline{TZ6}$ ).  $\overline{TZ1}$  to  $\overline{TZ3}$  are sourced from the GPIO mux.  $\overline{TZ4}$  is sourced from a combination of EQEP1ERR and EQEP2ERR signals.  $\overline{TZ5}$  is connected to the system oscillator or PLL clock fail logic, and  $\overline{TZ6}$  is sourced from the debug mode halt indication output from the CPU. These signals indicate fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when faults occur.


**Figure 18-35. Trip-Zone Submodule**

### 18.2.7.1 Purpose of the Trip-Zone Submodule

The key functions of the Trip-Zone submodule are:

- Trip inputs  $\overline{TZ1}$  to  $\overline{TZ6}$  are mapped to all ePWM modules.
- Upon a fault indication, either no action is taken or the ePWM outputs can be forced to one of the following:
  - High
  - Low
  - High-impedance
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Support for digital compare tripping (DC) based on state of on-chip analog comparator module outputs and/or  $\overline{TZ1}$  to  $\overline{TZ3}$  signals.
- Each trip-zone input and digital compare (DC) submodule DCAEVT1/2 or DCBEVT1/2 force event can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone input.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if it is not required.

### 18.2.7.2 Controlling and Monitoring the Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

**Table 18-18. Trip-Zone Submodule Registers**

Register Name	Address Offset	Shadowed	Description <sup>(2)</sup>
TZSEL	0x0026	No	Trip-Zone Select Register
TZDCSEL	0x0024	No	Trip-zone Digital Compare Select Register <sup>(1)</sup>
TZCTL	0x002A	No	Trip-Zone Control Register
TZEINT	0x0028	No	Trip-Zone Enable Interrupt Register
TZFLG	0x002E	No	Trip-Zone Flag Register
TZCLR	0x002C	No	Trip-Zone Clear Register
TZFRC	0x0032	No	Trip-Zone Force Register

(1) This register is discussed in more detail in [Section 18.2.9](#).

(2) All trip-zone registers are writable only in privileged mode.

### 18.2.7.3 Operational Highlights for the Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals  $\overline{TZ1}$  to  $\overline{TZ6}$  (also collectively referred to as  $\overline{TZn}$ ) are active low input signals. When one of these signals goes low, or when a DCAEVT1/2 or DCBEVT1/2 force happens based on the TZDCSEL register event selection, it indicates that a trip event has occurred. Each ePWM module can be individually configured to ignore or use each of the trip-zone signals or DC events. Which trip-zone signals or DC events are used by a particular ePWM module is determined by the TZSEL register for that specific ePWM module. The trip-zone signals may or may not be synchronized to the system clock (VCLK4) and digitally filtered within the GPIO MUX block. A minimum of 3\*TBCLK low pulse width on  $\overline{TZn}$  inputs is sufficient to trigger a fault condition on the ePWM module. If the pulse width is less than this, the trip condition may not be latched. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on  $\overline{TZn}$  inputs. The GPIOs or peripherals must be appropriately configured. For more information, see the IOMM chapter of the device technical reference manual.

Each  $\overline{TZn}$  input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for an ePWM module. DCAEVT1 and DCBEVT1 events can be configured to directly trip an ePWM module or provide a one-shot trip event to the module. Likewise, DCAEVT2 and DCBEVT2 events can also be configured to directly trip an ePWM module or provide a cycle-by-cycle trip event to the module. This configuration is determined by the TZSEL[DCAEVT1/2], TZSEL[DCBEVT1/2], TZSEL[CBCn], and TZSEL[OSHTn] control bits (where n corresponds to the trip input) respectively.

- **Cycle-by-Cycle (CBC):**

When a cycle-by-cycle trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the EPWMxA and/or EPWMxB output. [Table 18-19](#) lists the possible actions. In addition, the cycle-by-cycle trip event flag (TZFLG[CBC]) is set and a EPWMx\_TZINT interrupt is generated if it is enabled in the TZEINT register and VIM peripheral.

If the CBC interrupt is enabled via the TZEINT register, and DCAEVT2 or DCBEVT2 are selected as CBC trip sources via the TZSEL register, it is not necessary to also enable the DCAEVT2 or DCBEVT2 interrupts in the TZEINT register, as the DC events trigger interrupts through the CBC mechanism.

The specified condition on the inputs is automatically cleared when the ePWM time-base counter reaches zero (TBCTR = 0x0000) if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The TZFLG[CBC] flag bit will remain set until it is manually cleared by writing to the TZCLR[CBC] bit. If the cycle-by-cycle trip event is still present when the TZFLG[CBC] bit is cleared, then it will again be immediately set.

- **One-Shot (OSHT):**

When a one-shot trip event occurs, the action specified in the TZCTL[TZA] and TZCTL[TZB] bits is carried out immediately on the EPWMxA and/or EPWMxB output. [Table 18-19](#) lists the possible actions. In addition, the one-shot trip event flag (TZFLG[OST]) is set and a EPWMx\_TZINT interrupt is generated if it is enabled in the TZEINT register and VIM peripheral. The one-shot trip condition must be cleared manually by writing to the TZCLR[OST] bit.

If the one-shot interrupt is enabled via the TZEINT register, and DCAEVT1 or DCBEVT1 are selected as OSHT trip sources via the TZSEL register, it is not necessary to also enable the DCAEVT1 or DCBEVT1 interrupts in the TZEINT register, as the DC events trigger interrupts through the OSHT mechanism.

- **Digital Compare Events (DCAEVT1/2 and DCBEVT1/2):**

A digital compare DCAEVT1/2 or DCBEVT1/2 event is generated based on a combination of the DCAH/DCAL and DCBH/DCBL signals as selected by the TZDZSEL register. The signals which source the DCAH/DCAL and DCBH/DCBL signals are selected via the DCTRISEL register and can be either trip zone input pins. For more information on the digital compare submodule signals, see [Section 18.2.9](#).

When a digital compare event occurs, the action specified in the TZCTL[DCAEVT1/2] and TZCTL[DCBEVT1/2] bits is carried out immediately on the EPWMxA and/or EPWMxB output. [Table 18-19](#) lists the possible actions. In addition, the relevant DC trip event flag (TZFLG[DCAEVT1/2] / TZFLG[DCBEVT1/2]) is set and a EPWMx\_TZINT interrupt is generated if it is enabled in the TZEINT register and VIM peripheral.

The specified condition on the pins is automatically cleared when the DC trip event is no longer present. The TZFLG[DCAEVT1/2] or TZFLG[DCBEVT1/2] flag bit will remain set until it is manually cleared by writing to the TZCLR[DCAEVT1/2] or TZCLR[DCBEVT1/2] bit. If the DC trip event is still present when the TZFLG[DCAEVT1/2] or TZFLG[DCBEVT1/2] flag is cleared, then it will again be immediately set.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the TZCTL register bit fields. One of four possible actions, shown in [Table 18-19](#), can be taken on a trip event.

**Table 18-19. Possible Actions On a Trip Event**

TZCTL Register bit-field Settings	EPWMxA and/or EPWMxB	Comment
0,0	High-Impedance	Tripped
0,1	Force to High State	Tripped
1,0	Force to Low State	Tripped
1,1	No Change	Do Nothing. No change is made to the output.

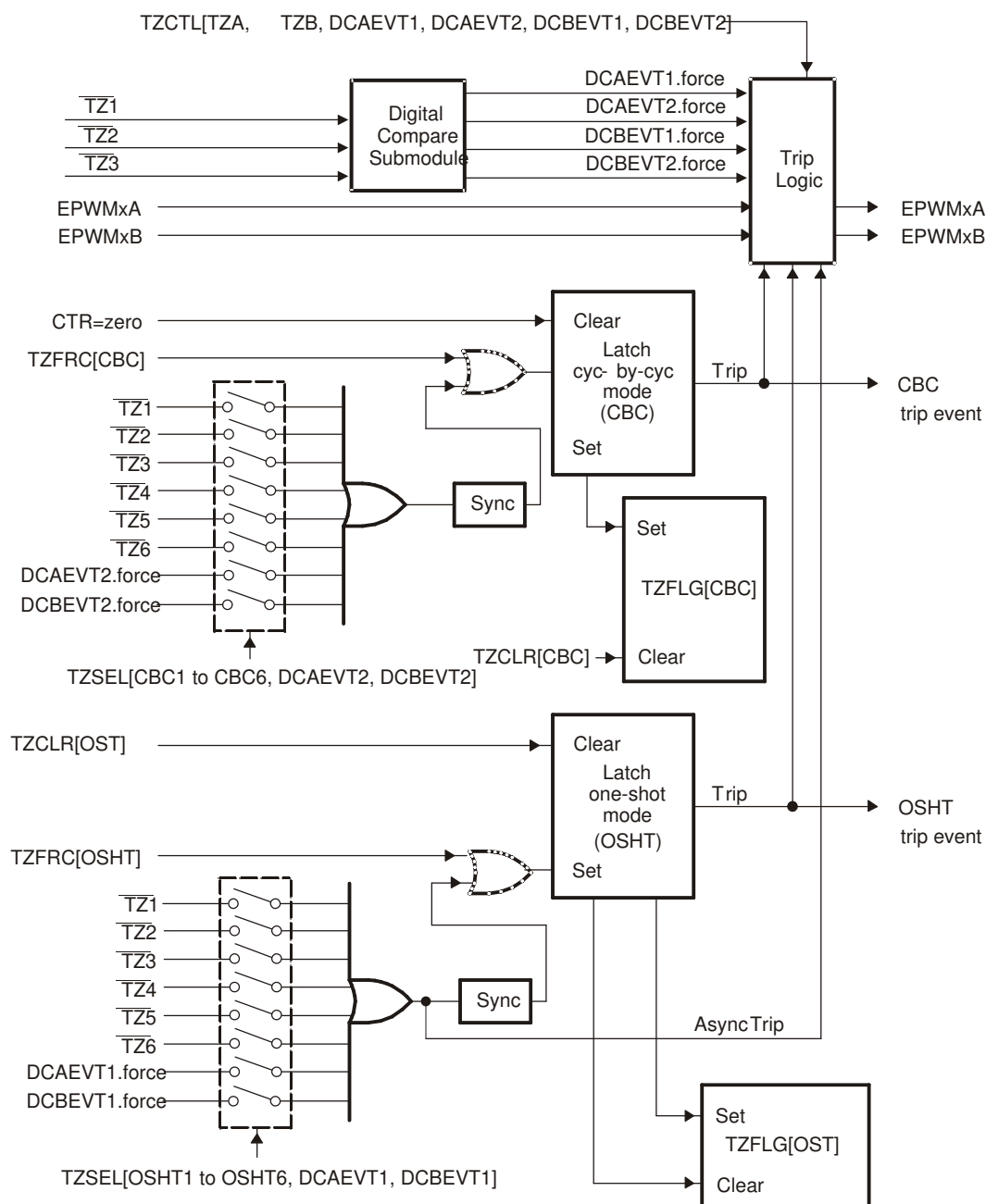
---

**Example 18-7. Trip-Zone Configurations**

---

### 18.2.7.4 Generating Trip Event Interrupts

Figure 18-36 and Figure 18-37 illustrate the trip-zone submodule control and interrupt logic, respectively. DCAEVT1/2 and DCBEVT1/2 signals are described in further detail in Section 18.2.9.



**Figure 18-36. Trip-Zone Submodule Mode Control Logic**



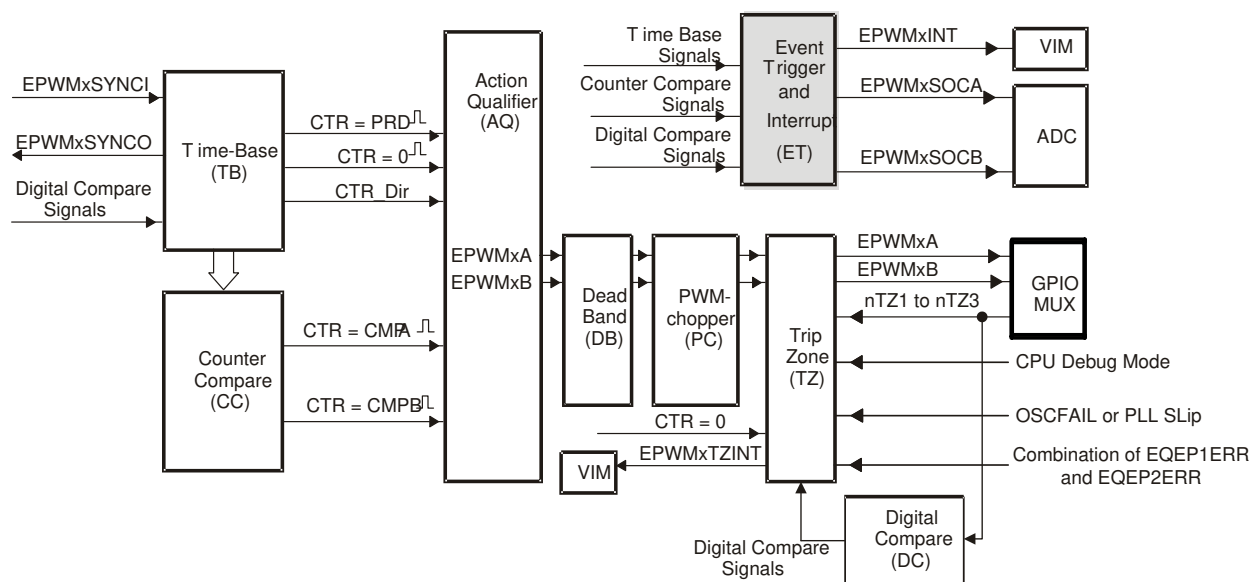


### 18.2.8 Event-Trigger (ET) Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base, counter-compare and digital-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests and ADC start of conversion at:
  - Every event
  - Every second event
  - Every third event
- Provides full visibility of event generation via event counters and flags
- Allows software forcing of Interrupts and ADC start of conversion

The event-trigger submodule manages the events generated by the time-base submodule, the counter-compare submodule, and the digital-compare submodule to generate an interrupt to the CPU and/or a start of conversion pulse to the ADC when a selected event occurs. Figure 18-38 illustrates where the event-trigger submodule fits within the ePWM system.



**Figure 18-38. Event-Trigger Submodule**

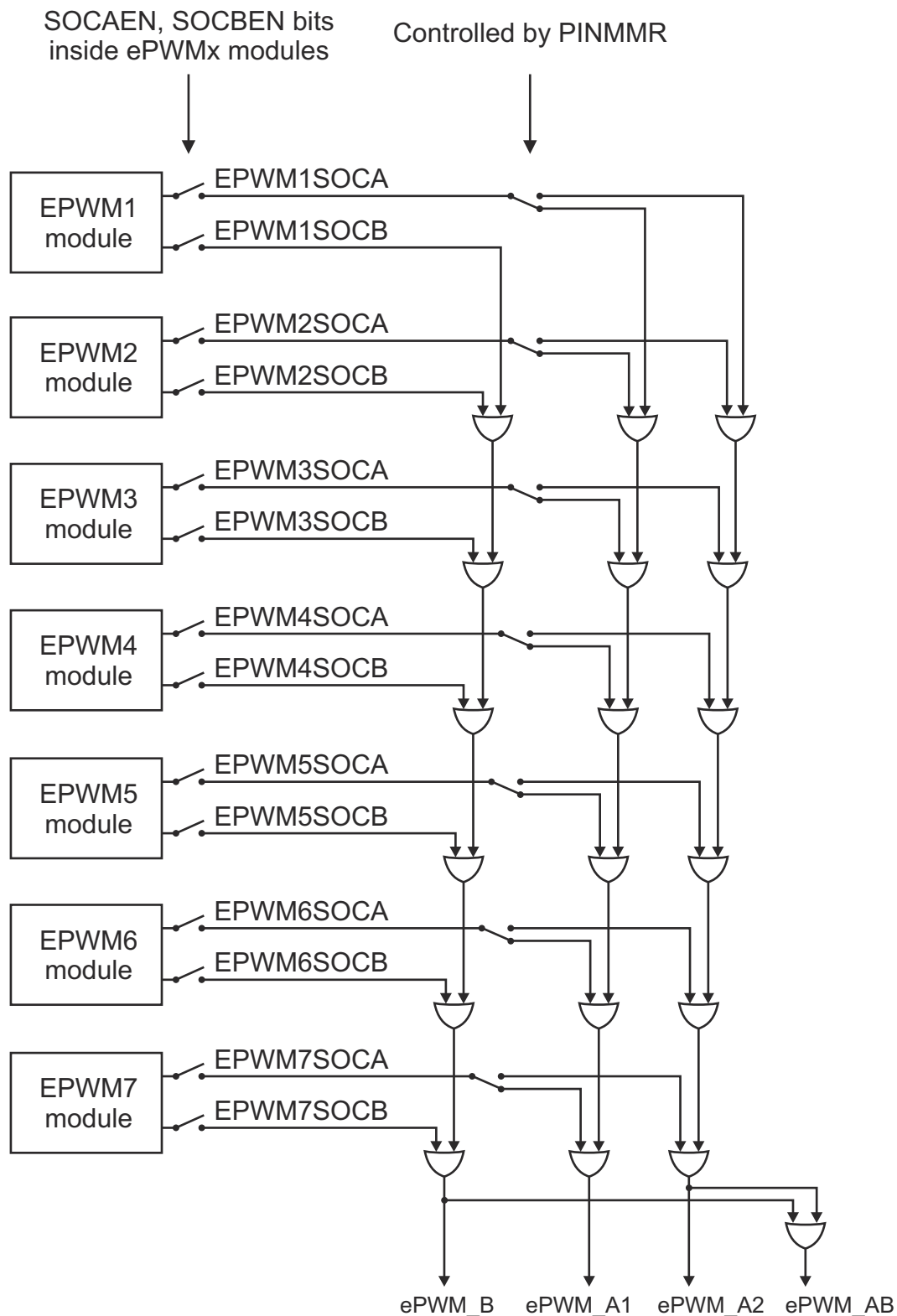
#### 18.2.8.1 Operational Overview of the Event-Trigger Submodule

The following sections describe the event-trigger submodule's operational highlights.

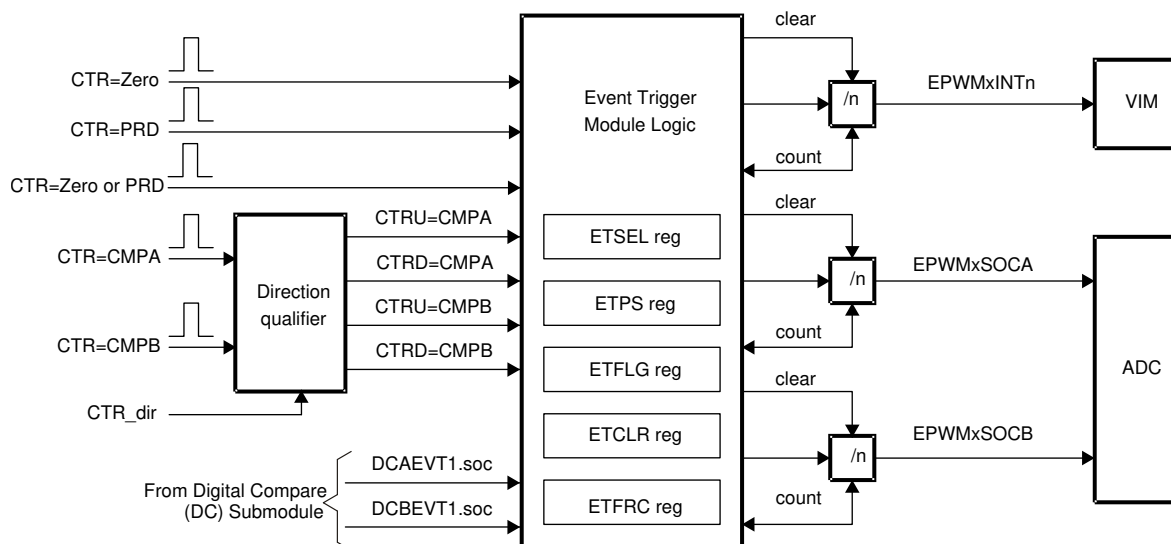
Each ePWM module has one interrupt request line connected to the VIM and two start of conversion signals connected to the ADC module. As shown in Figure 18-39, the ePWMxSOCA and ePWMxSOCB signals are combined to generate four special signals that can be used to trigger an ADC start of conversion, and hence multiple modules can initiate an ADC start of conversion via the ADC trigger inputs.

The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 18-40) and can be configured to prescale these events before issuing an Interrupt request or an ADC start of conversion. The event-trigger prescaling logic can issue Interrupt requests and ADC start of conversion at:

- Every event
- Every second event
- Every third event



**Figure 18-39. Event-Trigger Submodule Inter-Connectivity of ADC Start of Conversion**



**Figure 18-40. Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs**

The key registers used to configure the event-trigger submodule are shown in [Table 18-20](#).

**Table 18-20. Event-Trigger Submodule Registers**

Register Name	Address Offset	Shadowed	Description
ETSEL	0x0030	No	Event-trigger Selection Register
ETPS	0x0036	No	Event-trigger Prescale Register
ETFLG	0x0034	No	Event-trigger Flag Register
ETCLR	0x003A	No	Event-trigger Clear Register
ETFRC	0x0038	No	Event-trigger Force Register

- ETSEL—This selects which of the possible events will trigger an interrupt or start an ADC conversion
- ETPS—This programs the event prescaling options mentioned above.
- ETFLG—These are flag bits indicating status of the selected and prescaled events.
- ETCLR—These bits allow you to clear the flag bits in the ETFLG register via software.
- ETFRC—These bits allow software forcing of an event. Useful for debugging or s/w intervention.

A more detailed look at how the various register bits interact with the Interrupt and ADC start of conversion logic are shown in [Figure 18-41](#), [Figure 18-42](#), and [Figure 18-43](#).

[Figure 18-41](#) shows the event-trigger's interrupt generation logic. The interrupt-period (ETPS[INTPRD]) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt.
- Generate an interrupt on every event
- Generate an interrupt on every second event
- Generate an interrupt on every third event

Which event can cause an interrupt is configured by the interrupt selection (ETSEL[INTSEL]) bits. The event can be one of the following:

- Time-base counter equal to zero (TBCTR = 0x0000).
- Time-base counter equal to period (TBCTR = TBPRD).
- Time-base counter equal to zero or period (TBCTR = 0x0000 || TBCTR = TBPRD)
- Time-base counter equal to the compare A register (CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (CMPA) when the timer is decrementing.

- Time-base counter equal to the compare B register (CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (CMPB) when the timer is decrementing.

The number of events that have occurred can be read from the interrupt event counter (ETPS[INTCNT]) register bits. That is, when the specified event occurs the ETPS[INTCNT] bits are incremented until they reach the value specified by ETPS[INTPRD]. When ETPS[INTCNT] = ETPS[INTPRD] the counter stops counting and its output is set. The counter is only cleared when an interrupt is sent to the VIM.

When ETPS[INTCNT] reaches ETPS[INTPRD] the following behaviors will occur:

- If interrupts are enabled, ETSEL[INTEN] = 1 and the interrupt flag is clear, ETFLG[INT] = 0, then an interrupt pulse is generated and the interrupt flag is set, ETFLG[INT] = 1, and the event counter is cleared ETPS[INTCNT] = 0. The counter will begin counting events again.
- If interrupts are disabled, ETSEL[INTEN] = 0, or the interrupt flag is set, ETFLG[INT] = 1, the counter stops counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD].
- If interrupts are enabled, but the interrupt flag is already set, then the counter will hold its output high until the ETFLG[INT] flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing to the INTPRD bits will automatically clear the counter INTCNT = 0 and the counter output will be reset (so no interrupts are generated). Writing a 1 to the ETFRC[INT] bit will increment the event counter INTCNT. The counter will behave as described above when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events will be detected and the ETFRC[INT] bit is also ignored.

The above definition means that you can generate an interrupt on every event, on every second event, or on every third event. An interrupt cannot be generated on every fourth or more events.

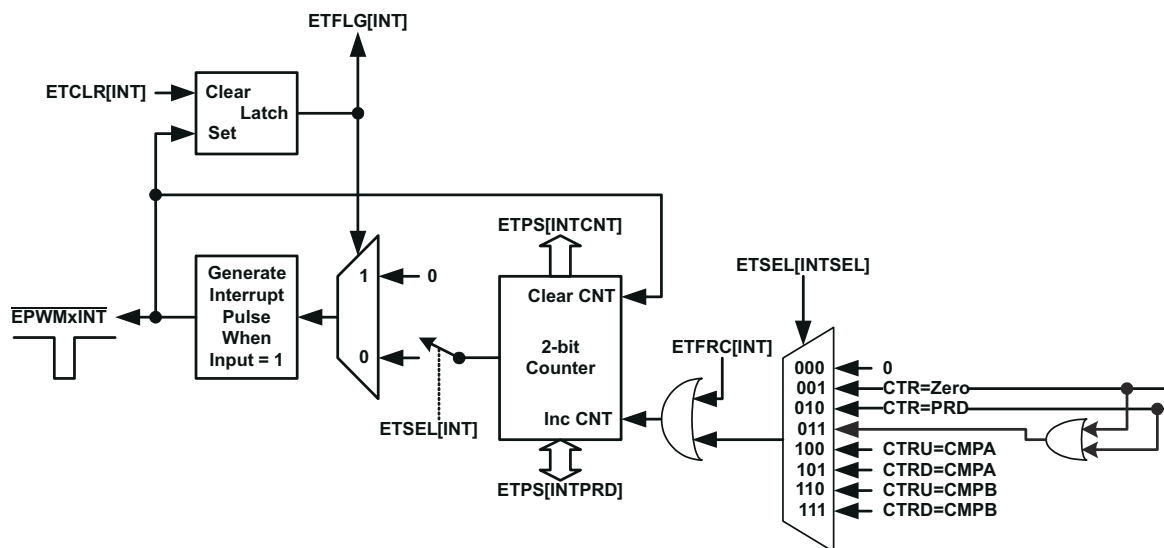
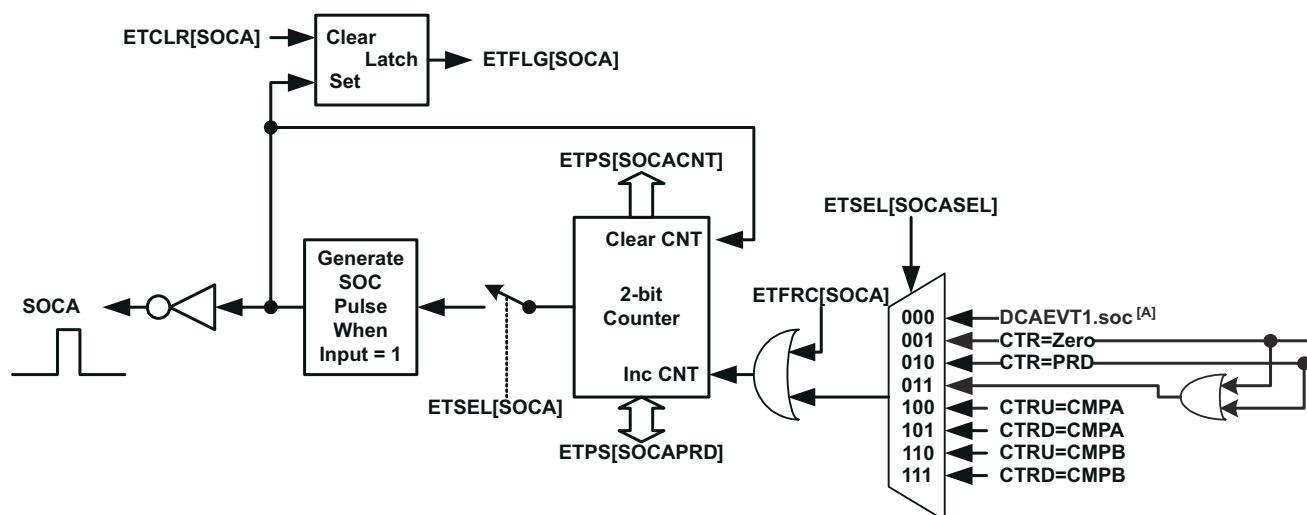


Figure 18-41. Event-Trigger Interrupt Generator

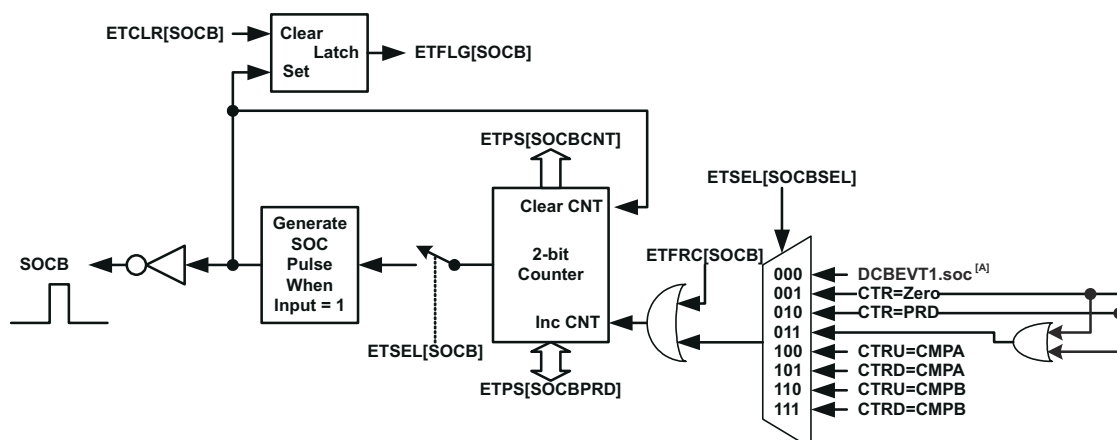
Figure 18-42 shows the operation of the event-trigger's start-of-conversion-A (SOCA) pulse generator. The ETPS[SOCACNT] counter and ETPS[SOCAPRD] period values behave similarly to the interrupt generator except that the pulses are continuously generated. That is, the pulse flag ETFLG[SOCA] is latched when a pulse is generated, but it does not stop further pulse generation. The enable/disable bit ETSEL[SOCAEN] stops pulse generation, but input events can still be counted until the period value is reached as with the interrupt generation logic. The event that will trigger an SOCA and SOCB pulse can be configured separately in the ETSEL[SOCASEL] and ETSEL[SOCBSEL] bits. The possible events are the same events that can be specified for the interrupt generation logic with the addition of the DCAEVT1.soc and DCBEVT1.soc event signals from the digital compare (DC) submodule.



A. The DCAEVT1.soc signals are signals generated by the Digital compare (DC) submodule described later in [Section 18.2.9](#).

**Figure 18-42. Event-Trigger SOCA Pulse Generator**

Figure 18-43 shows the operation of the event-trigger's start-of-conversion-B (SOCB) pulse generator. The event-trigger's SOCB pulse generator operates the same way as the SOCA.



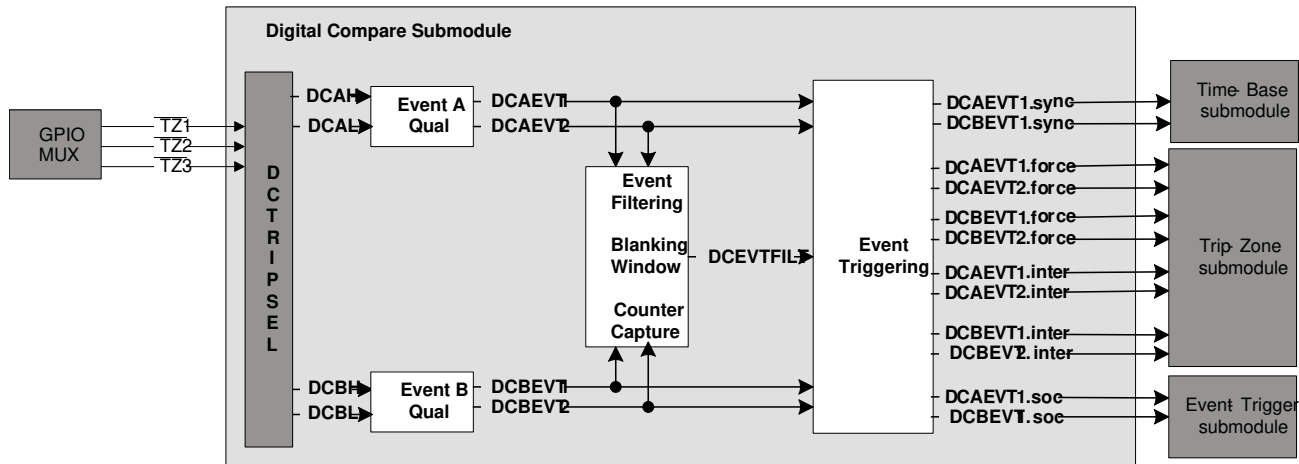
A. The DCBEVT1.soc signals are signals generated by the Digital compare (DC) submodule described later in [Section 18.2.9](#).

**Figure 18-43. Event-Trigger SOCB Pulse Generator**

### 18.2.9 Digital Compare (DC) Submodule

Figure 18-44 illustrates where the digital compare (DC) submodule signals interface to other submodules in the ePWM system.

The digital compare (DC) submodule compares signals external to the ePWM module to directly generate PWM events/actions that then feed to the event-trigger, trip-zone, and time-base submodules. Additionally, blanking window functionality is supported to filter noise or unwanted pulses from the DC event signals.



**Figure 18-44. Digital-Compare Submodule High-Level Block Diagram**

#### 18.2.9.1 Purpose of the Digital Compare Submodule

The key functions of the digital compare submodule are:

- TZ1, TZ2, and TZ3 inputs generate Digital Compare A High/Low (DCAH, DCAL) and Digital Compare B High/Low (DCBH, DCBL) signals.
- DCAH/L and DCBH/L signals trigger events which can then either be filtered or fed directly to the trip-zone, event-trigger, and time-base submodules to:
  - generate a trip zone interrupt
  - generate an ADC start of conversion
  - force an event
  - generate a synchronization event for synchronizing the ePWM module TBCTR.
- Event filtering (blanking window logic) can optionally blank the input signal to remove noise.

### 18.2.9.2 Controlling and Monitoring the Digital Compare Submodule

The digital compare submodule operation is controlled and monitored through the following registers:

**Table 18-21. Digital Compare Submodule Registers**

Register Name	Address Offset	Shadowed	Description
TZDCSEL <sup>(1) (2)</sup>	0x0024	No	Trip Zone Digital Compare Select Register
DCTRIPSEL <sup>(1)</sup>	0x0062	No	Digital Compare Trip Select Register
DCACTL <sup>(1)</sup>	0x0060	No	Digital Compare A Control Register
DCBCTL <sup>(1)</sup>	0x0066	No	Digital Compare B Control Register
DCFCTL <sup>(1)</sup>	0x0064	No	Digital Compare Filter Control Register
DCCAPCTL <sup>(1)</sup>	0x006A	No	Digital Compare Capture Control Register
DCOFFSET	0x0068	Writes	Digital Compare Filter Offset Register
DCOFFSETCNT	0x006E	No	Digital Compare Filter Offset Counter Register
DCFWINDOW	0x006C	No	Digital Compare Filter Window Register
DCFWINDOWCNT	0x0072	No	Digital Compare Filter Window Counter Register
DCCAP	0x0070	Yes	Digital Compare Counter Capture Register

(1) These registers are writable only in privileged mode.

(2) The TZDCSEL register is part of the trip-zone submodule but is mentioned again here because of its functional significance to the digital compare submodule.

### 18.2.9.3 Operation Highlights of the Digital Compare Submodule

The following sections describe the operational highlights and configuration options for the digital compare submodule.

#### 18.2.9.3.1 Digital Compare Events

As illustrated in [Figure 18-44](#), trip zone inputs ( $\overline{TZ1}$ ,  $\overline{TZ2}$ , and  $\overline{TZ3}$ ) can be selected via the DCTRIPSEL bits to generate the Digital Compare A High and Low (DCAH/L) and Digital Compare B High and Low (DCBH/L) signals. Then, the configuration of the TZDCSEL register qualifies the actions on the selected DCAH/L and DCBH/L signals, which generate the DCAEVT1/2 and DCBEVT1/2 events (Event Qualification A and B).

#### Note

The  $\overline{TZn}$  signals, when used as a DCEVT tripping functions, are treated as a normal input signal and can be defined to be active high or active low inputs. EPWM outputs are asynchronously tripped when either the  $\overline{TZn}$ , DCAEVTx.force, or DCBEVTx.force signals are active. For the condition to remain latched, a minimum of  $3 \times TBCLK$  sync pulse width is required. If pulse width is  $< 3 \times TBCLK$  sync pulse width, the trip condition may or may not get latched by CBC or OST latches.



The DCAEVT1/2 and DCBEVT1/2 events can then be filtered to provide a filtered version of the event signals (DCEVTFILT) or the filtering can be bypassed. Filtering is discussed further in [Section 18.2.9.3.2](#). Either the DCAEVT1/2 and DCBEVT1/2 event signals or the filtered DCEVTFILT event signals can generate a force to the trip zone module, a TZ interrupt, an ADC SOC, or a PWM sync signal.

- **force signal:**

DCAEVT1/2.force signals force trip zone conditions which either directly influence the output on the EPWMxA pin (via TZCTL[DCAEVT1 or DCAEVT2] configurations) or, if the DCAEVT1/2 signals are selected as one-shot or cycle-by-cycle trip sources (via the TZSEL register), the DCAEVT1/2.force signals can effect the trip action via the TZCTL[TZA] configuration. The DCBEVT1/2.force signals behaves similarly, but affect the EPWMxB output pin instead of the EPWMxA output pin.

The priority of conflicting actions on the TZCTL register is as follows (highest priority overrides lower priority):

Output EPWMxA: TZA (highest) -> DCAEVT1 -> DCAEVT2 (lowest)

Output EPWMxB: TzB (highest) -> DCBEVT1 -> DCBEVT2 (lowest)

- **interrupt signal:**

DCAEVT1/2.interrupt signals generate trip zone interrupts to the VIM. To enable the interrupt, the user must set the DCAEVT1, DCAEVT2, DCBEVT1, or DCBEVT2 bits in the TZEINT register. Once one of these events occurs, an EPWMxTZINT interrupt is triggered, and the corresponding bit in the TZCLR register must be set in order to clear the interrupt.

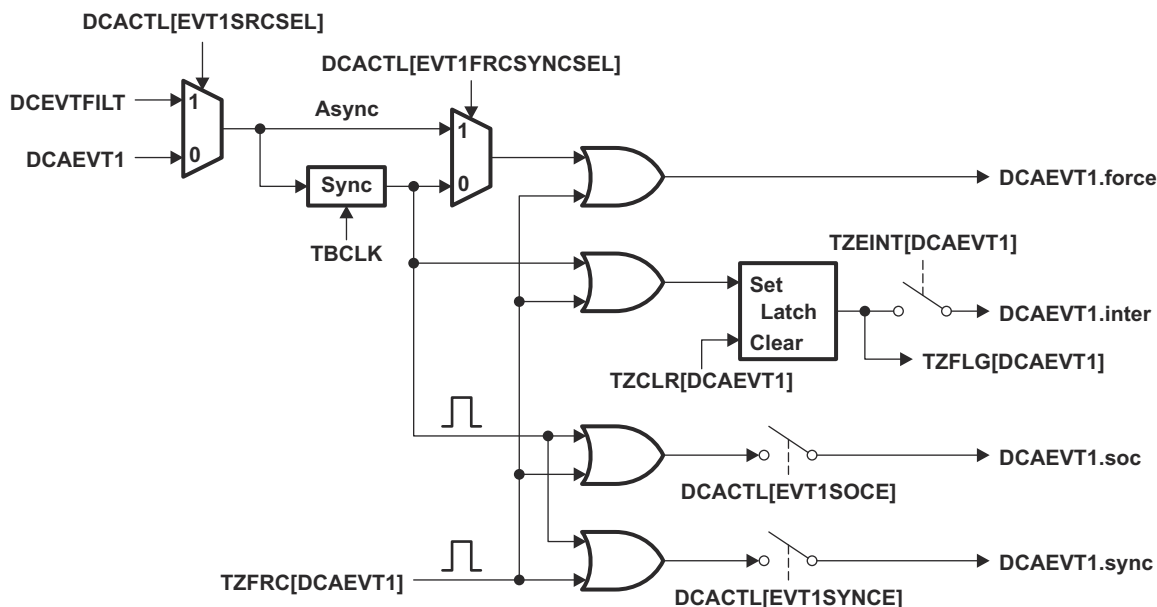
- **soc signal:**

The DCAEVT1.soc signal interfaces with the event-trigger submodule and can be selected as an event which generates an ADC start-of-conversion-A (SOCA) pulse via the ETSEL[SOCASEL] bit. Likewise, the DCBEVT1.soc signal can be selected as an event which generates an ADC start-of-conversion-B (SOCB) pulse via the ETSEL[SOCBSEL] bit.

- **sync signal:**

The DCAEVT1.sync and DCBEVT1.sync events are ORed with the EPWMxSYNCl input signal and the TBCTL[SWFSYNc] signal to generate a synchronization pulse to the time-base counter.

[Figure 18-45](#) and [Figure 18-46](#) show how the DCAEVT1, DCAEVT2, or DCEVTFILT signals are processed to generate the digital compare A event force, interrupt, soc and sync signals.



**Figure 18-45. DCAEVT1 Event Triggering**

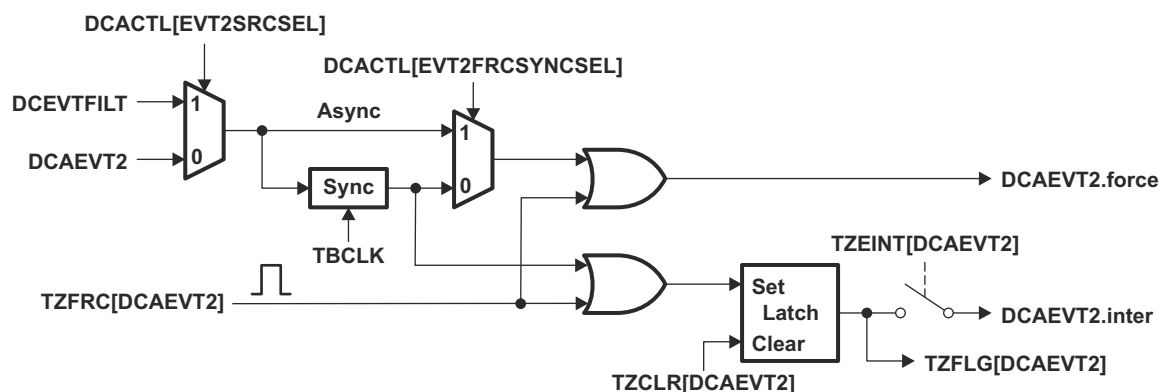
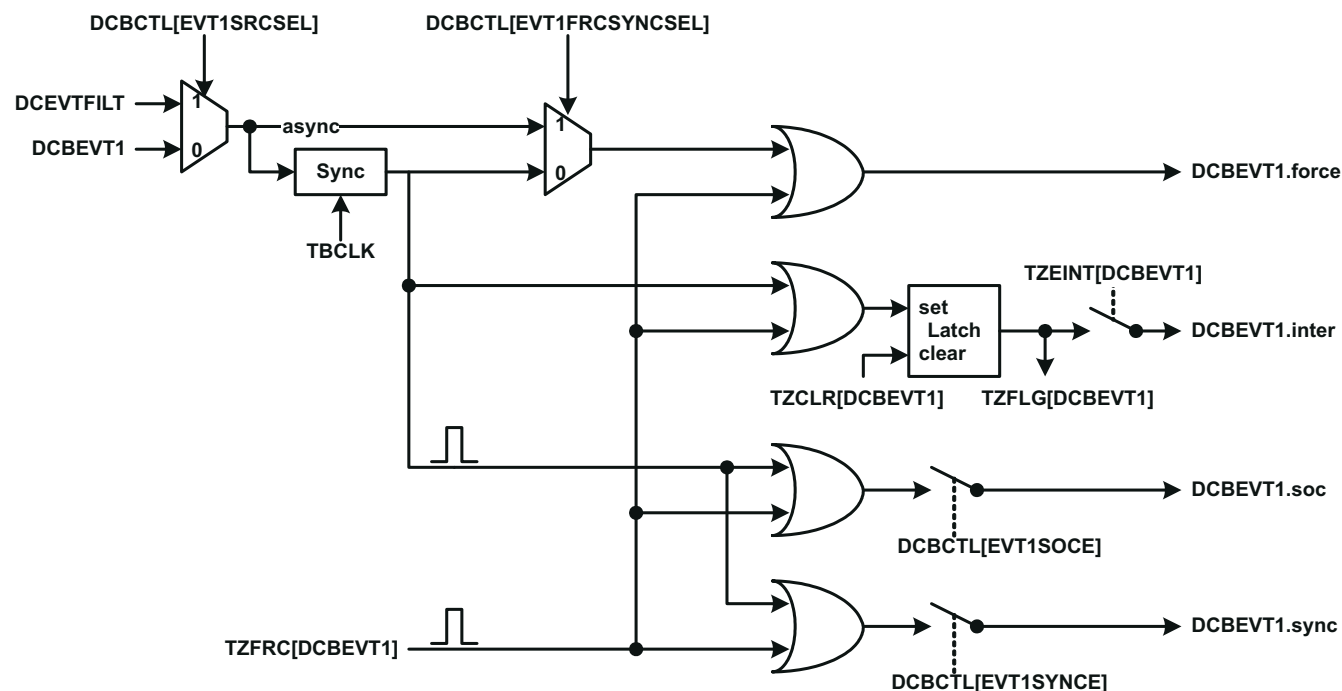

**Figure 18-46. DCAEVT2 Event Triggering**

Figure 18-47 and Figure 18-48 show how the DCBEVT1, DCBEVT2, or DCEVTFILT signals are processed to generate the digital compare B event force, interrupt, soc, and sync signals.


**Figure 18-47. DCBEVT1 Event Triggering**

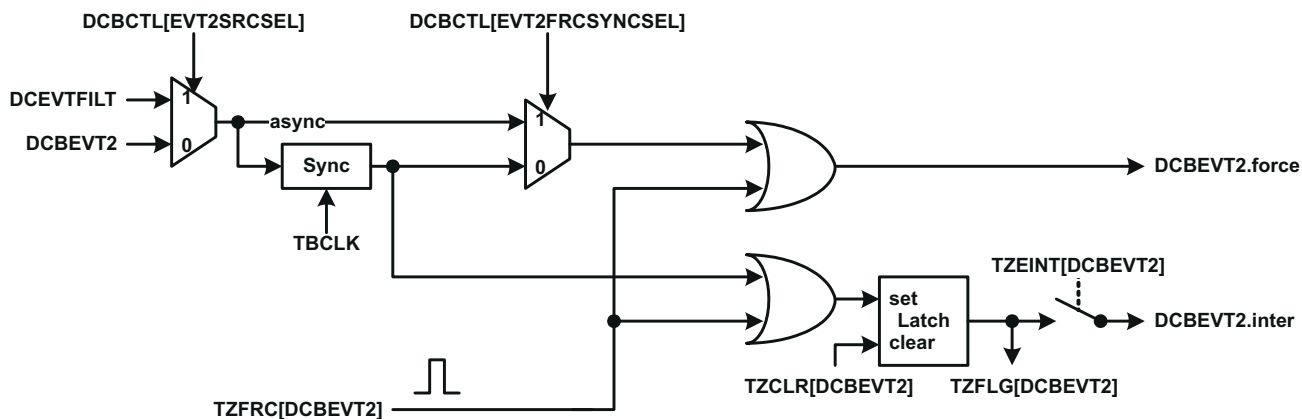


Figure 18-48. DCBEVT2 Event Triggering

### 18.2.9.3.2 Event Filtering

The DCAEVT1/2 and DCBEVT1/2 events can be filtered via event filtering logic to remove noise by optionally blanking events for a certain period of time. This is useful for cases where the analog comparator outputs may be selected to trigger DCAEVT1/2 and DCBEVT1/2 events, and the blanking logic is used to filter out potential noise on the signal prior to tripping the PWM outputs or generating an interrupt or ADC start-of-conversion. The event filtering can also capture the TBCTR value of the trip event. [Figure 18-49](#) shows the details of the event filtering logic.

If the blanking logic is enabled, one of the digital compare events – DCAEVT1, DCAEVT2, DCBEVT1, DCBEVT2 – is selected for filtering. The blanking window, which filters out all event occurrences on the signal while it is active, will be aligned to either a CTR = PRD pulse or a CTR = 0 pulse (configured by the DCFCTL[PULSESEL] bits). An offset value in TBCLK counts is programmed into the DCFOFFSET register, which determines at what point after the CTR = PRD or CTR = 0 pulse the blanking window starts. The duration of the blanking window, in number of TBCLK counts after the offset counter expires, is written to the DCFWINDOW register by the application. During the blanking window, all events are ignored. Before and after the blanking window ends, events can generate soc, sync, interrupt, and force signals as before.

[Figure 18-50](#) illustrates several timing conditions for the offset and blanking window within an ePWM period. Notice that if the blanking window crosses the CTR = 0 or CTR = PRD boundary, the next window still starts at the same offset value after the CTR = 0 or CTR = PRD pulse.

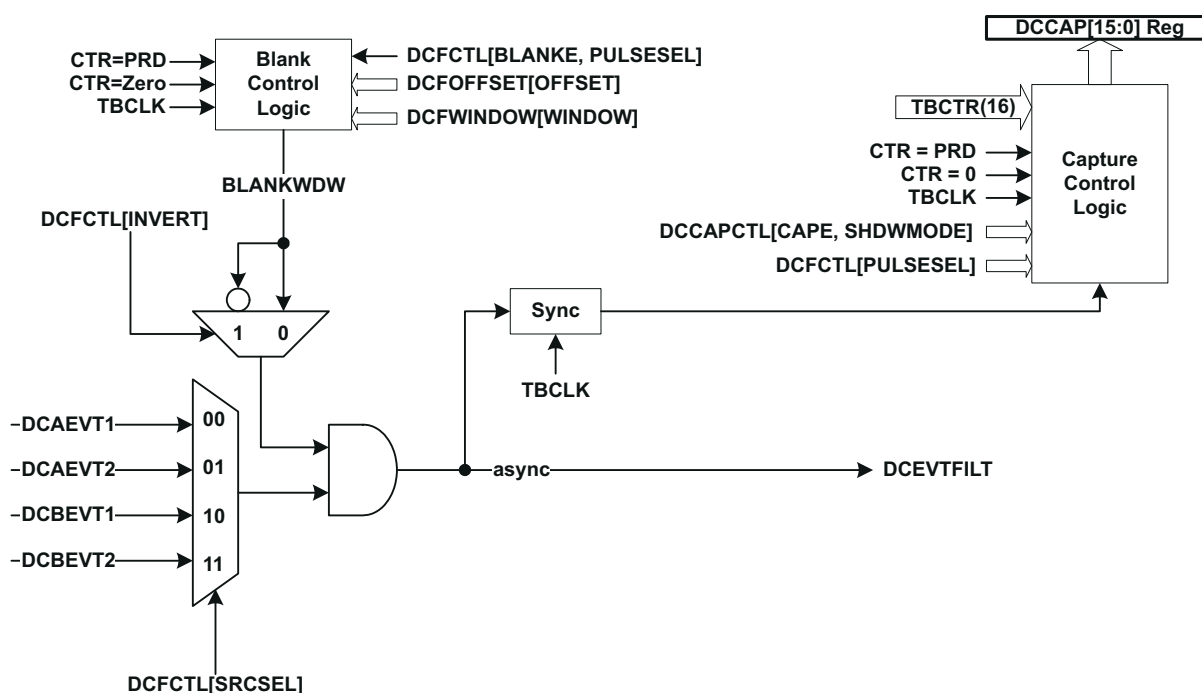


Figure 18-49. Event Filtering

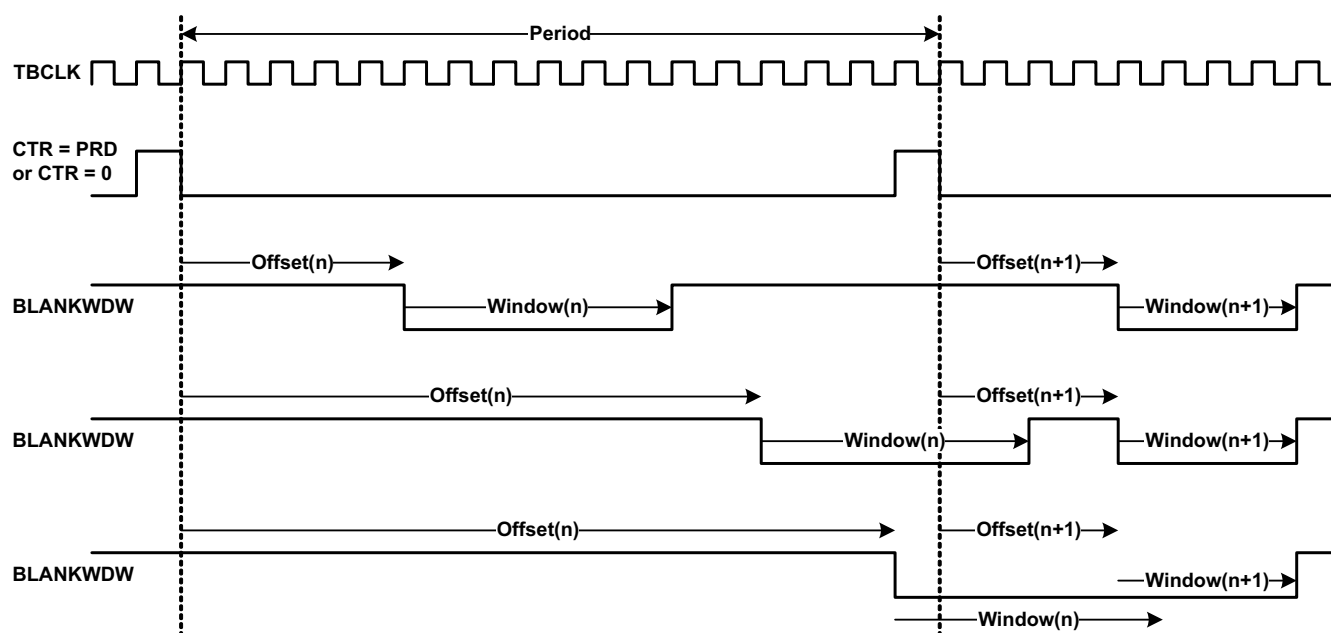


Figure 18-50. Blanking Window Timing Diagram

### 18.2.10 Proper Interrupt Initialization Procedure

When the ePWM peripheral clock is enabled it may be possible that interrupt flags may be set due to spurious events due to the ePWM registers not being properly initialized. The proper procedure for initializing the ePWM peripheral is as follows:

1. Disable global interrupts (CPU INTM flag)
2. Disable ePWM interrupts
3. Set TBCLKSYNC = 0
4. Initialize peripheral registers
5. Set TBCLKSYNC = 1
6. Clear any spurious ePWM flags (including interrupt flags)
7. Enable ePWM interrupts
8. Enable global interrupts

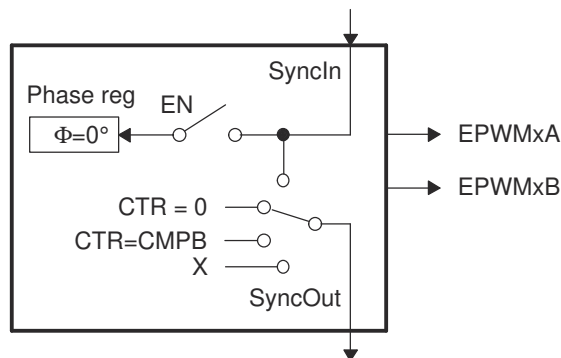
### 18.2.11

## 18.3 Application Examples

An ePWM module has all the local resources necessary to operate completely as a standalone module or to operate in synchronization with other identical ePWM modules.

### 18.3.1 Overview of Multiple Modules

Previously in this chapter, all discussions have described the operation of a single module. To facilitate the understanding of multiple modules working together in a system, the ePWM module described in reference is represented by the more simplified block diagram shown in [Figure 18-51](#). This simplified ePWM block shows only the key resources needed to explain how a multiswitch power topology is controlled with multiple ePWM modules working together.



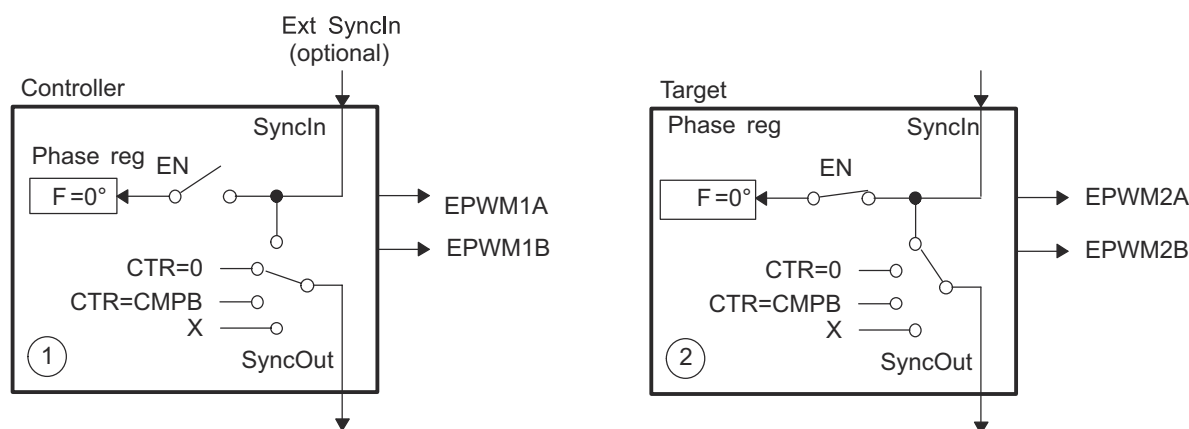
**Figure 18-51. Simplified ePWM Module**

### 18.3.2 Key Configuration Capabilities

The key configuration choices available to each module are as follows:

- Options for SyncIn
  - Load own counter with phase register on an incoming sync strobe—enable (EN) switch closed
  - Do nothing or ignore incoming sync strobe—enable switch open
  - Sync flow-through - SyncOut connected to SyncIn
  - Controller mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Controller mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)
- Options for SyncOut
  - Sync flow-through - SyncOut connected to SyncIn
  - Controller mode, provides a sync at PWM boundaries—SyncOut connected to CTR = PRD
  - Controller mode, provides a sync at any programmable point in time—SyncOut connected to CTR = CMPB
  - Module is in standalone mode and provides No sync to other modules—SyncOut connected to X (disabled)

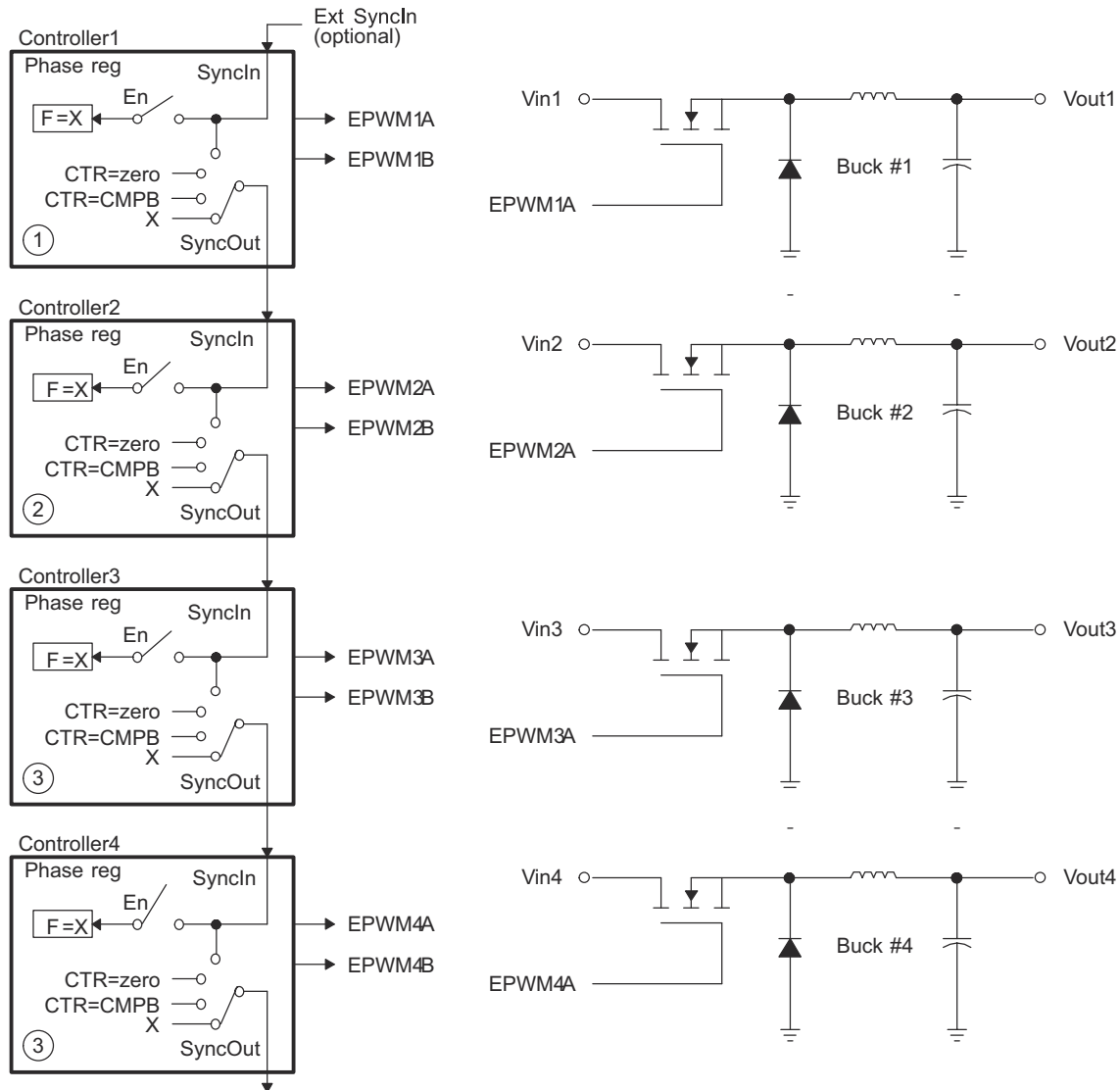
For each choice of SyncOut, a module may also choose to load its own counter with a new phase value on a SyncIn strobe input or choose to ignore it, that is, via the enable switch. Although various combinations are possible, the two most common—controller module and target module modes—are shown in [Figure 18-52](#).



**Figure 18-52. EPWM1 Configured as a Typical Controller, EPWM2 Configured as a Target**

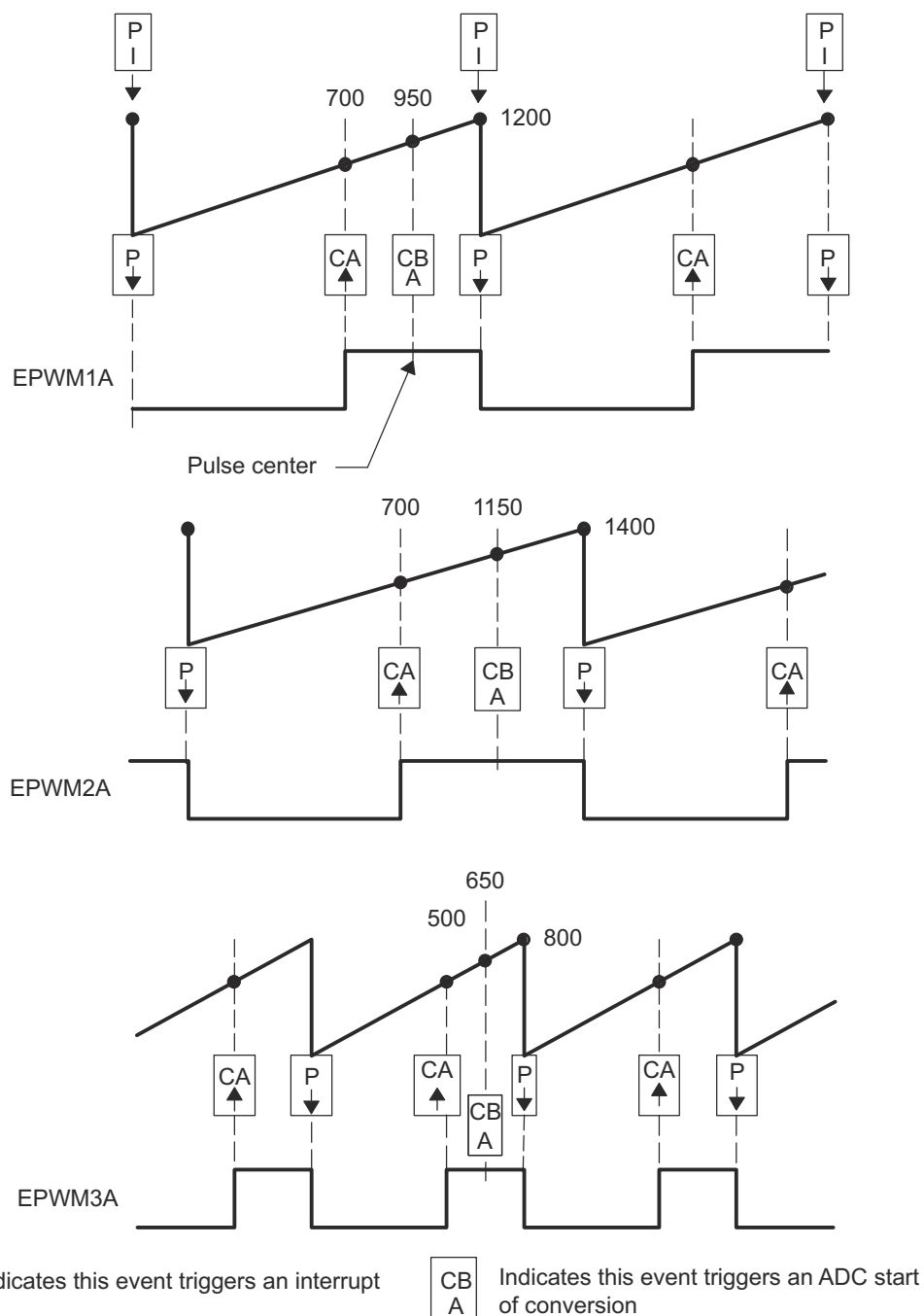
### 18.3.3 Controlling Multiple Buck Converters With Independent Frequencies

One of the simplest power converter topologies is the buck. A single ePWM module configured as a controller can control two buck stages with the same PWM frequency. If independent frequency control is required for each buck converter, then one ePWM module must be allocated for each converter stage. Figure 18-53 shows four buck stages, each running at independent frequencies. In this case, all four ePWM modules are configured as Controllers and no synchronization is used. Figure 18-54 shows the waveforms generated by the setup shown in Figure 18-53; note that only three waveforms are shown, although there are four stages.



A.  $\odot = X$  indicates value in phase register is a "don't care"

**Figure 18-53. Control of Four Buck Stages. Here  $F_{PWM1} \neq F_{PWM2} \neq F_{PWM3} \neq F_{PWM4}$**



**Figure 18-54. Buck Waveforms for Figure 18-53 (Note: Only three bucks shown here)**

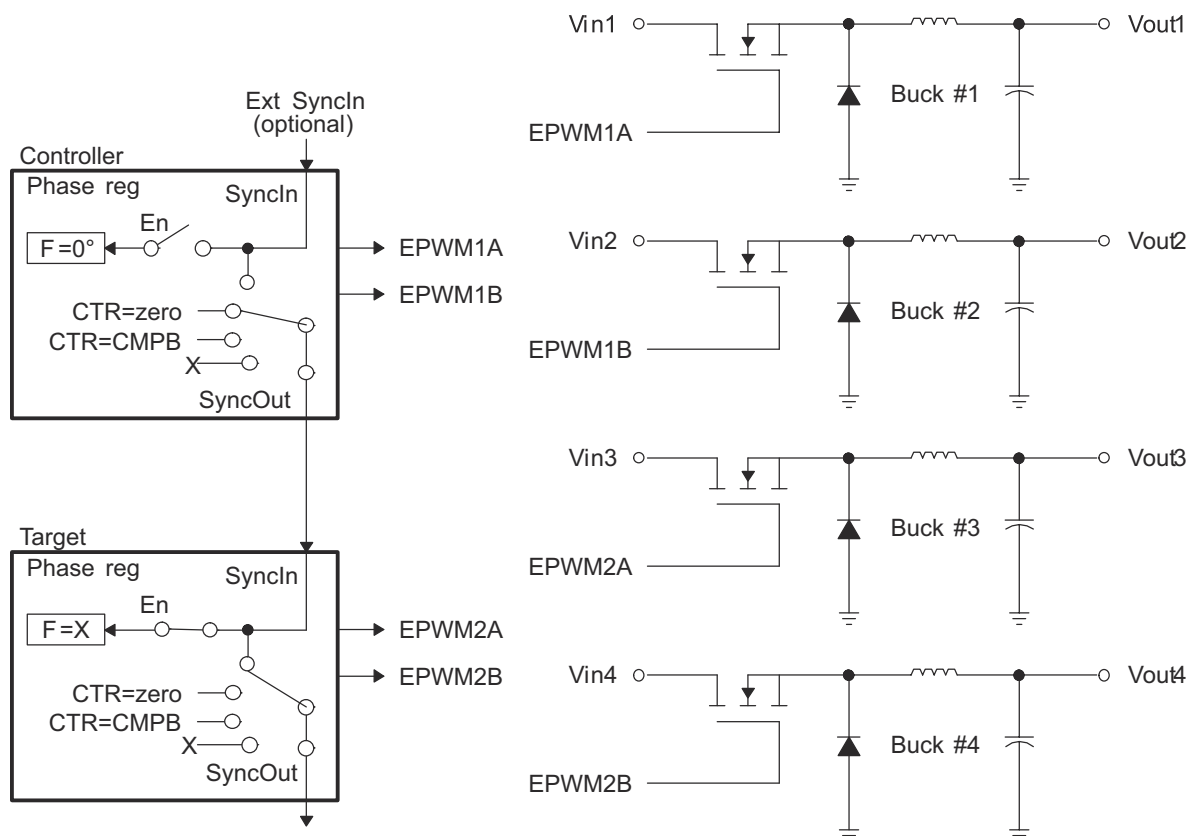


**Example 18-8. Configuration for Example in Figure 18-54**

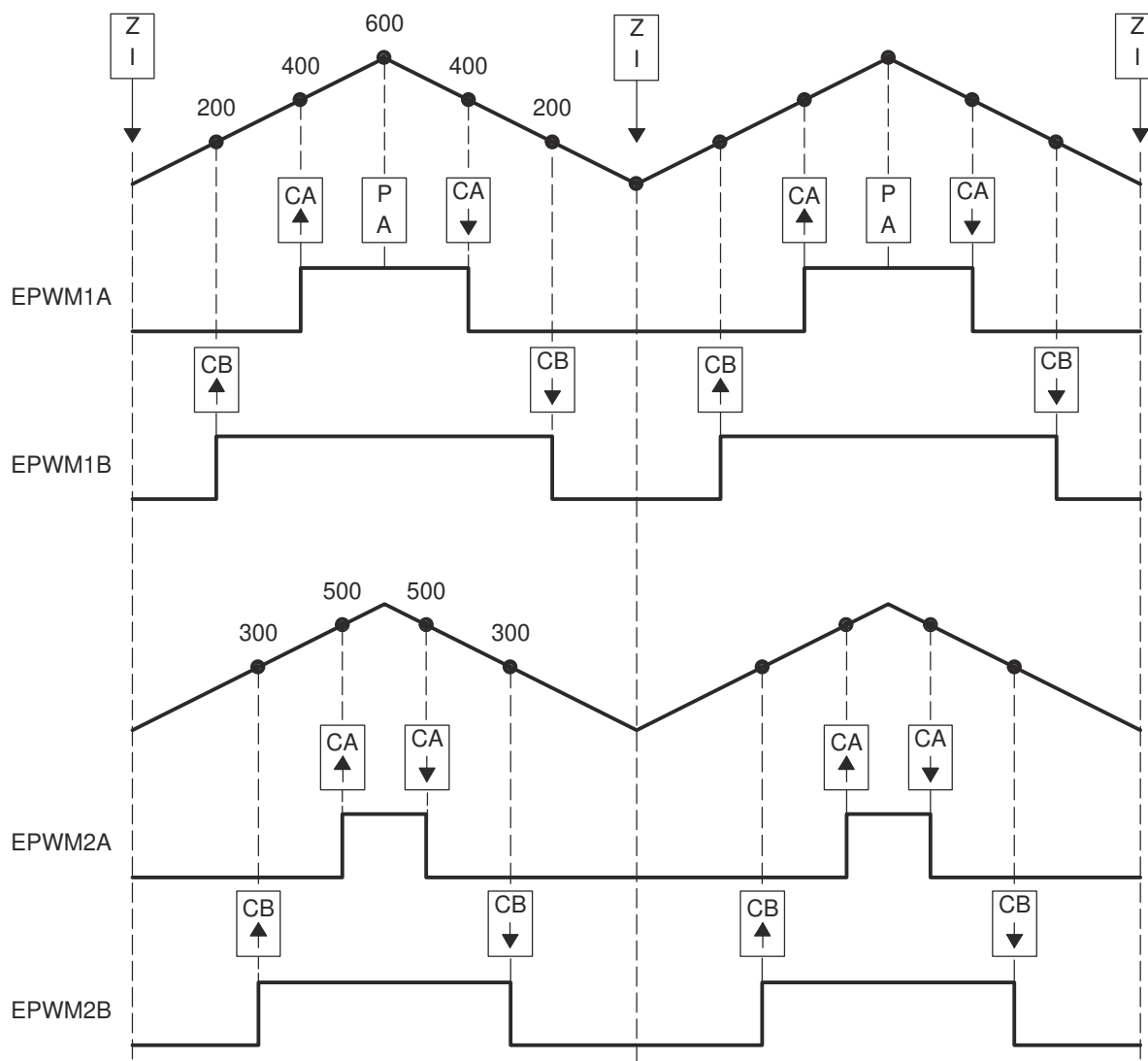
```
//=====
// (Note: code for only 3 modules shown)
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 1200; // Period = 1201 TBCLK counts
EPwm1Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Asymmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.PR = AQ_CLEAR;
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET;
// EPWM Module 2 config
EPwm2Regs.TBPRD = 1400; // Period = 1401 TBCLK counts
EPwm2Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP; // Asymmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.PR = AQ_CLEAR;
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET;
// EPWM Module 3 config
EPwm3Regs.TBPRD = 800; // Period = 801 TBCLK counts
EPwm3Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm3Regs.TBCTL.bit.CTRMODE = TB_COUNT_UP;
EPwm3Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Phase loading disabled
EPwm3Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm3Regs.TBCTL.bit.SYNCSEL = TB_SYNC_DISABLE;
EPwm3Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm3Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm3Regs.AQCTLA.bit.PR = AQ_CLEAR;
EPwm3Regs.AQCTLA.bit.CAU = AQ_SET;
//
// Run Time (Note: Example execution of one run-time instant)
//=====
EPwm1Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM1A
EPwm2Regs.CMPA.half.CMPA = 700; // adjust duty for output EPWM2A
EPwm3Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM3A
```

### 18.3.4 Controlling Multiple Buck Converters With Same Frequencies

If synchronization is a requirement, ePWM module 2 can be configured as a target and can operate at integer multiple (N) frequencies of module 1. The sync signal from controller to target ensures these modules remain locked. Figure 18-55 shows such a configuration; Figure 18-56 shows the waveforms generated by the configuration.



**Figure 18-55. Control of Four Buck Stages. (Note:  $F_{PWM2} = N \times F_{PWM1}$ )**



**Figure 18-56. Buck Waveforms for Figure 18-55 (Note:  $F_{PWM2} = F_{PWM1}$ )**

**Example 18-9. Code Snippet for Configuration in Figure 18-55**

```
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm1Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Controller module
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.CBU = AQ_SET; // set actions for EPWM1B
EPwm1Regs.AQCTLB.bit.CBD = AQ_CLEAR;
// EPWM Module 2 config
EPwm2Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm2Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Target module
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.CAU = AQ_SET; // set actions for EPWM2A
EPwm2Regs.AQCTLA.bit.CAD = AQ_CLEAR;
EPwm2Regs.AQCTLB.bit.CBU = AQ_SET; // set actions for EPWM2B
EPwm2Regs.AQCTLB.bit.CBD = AQ_CLEAR;
//
// Run Time (Note: Example execution of one run-time instance)
//=====
EPwm1Regs.CMPA.half.CMPA = 400; // adjust duty for output EPWM1A
EPwm1Regs.CMPB = 200; // adjust duty for output EPWM1B
EPwm2Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM2A
EPwm2Regs.CMPB = 300; // adjust duty for output EPWM2B
```



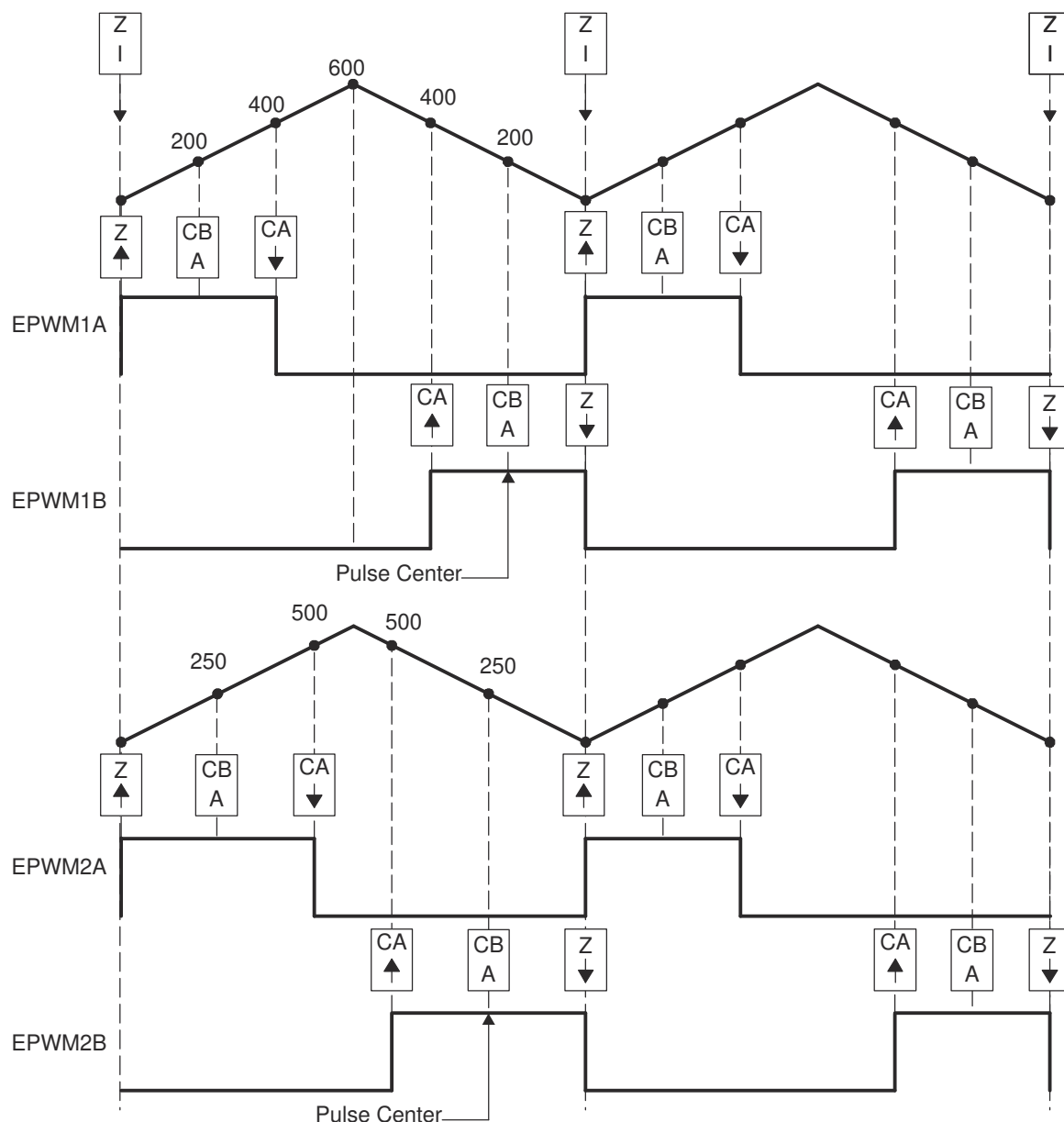


Figure 18-58. Half-H Bridge Waveforms for Figure 18-57 (Note: Here  $F_{PWM2} = F_{PWM1}$  )

### Example 18-10. Code Snippet for Configuration in Figure 18-57

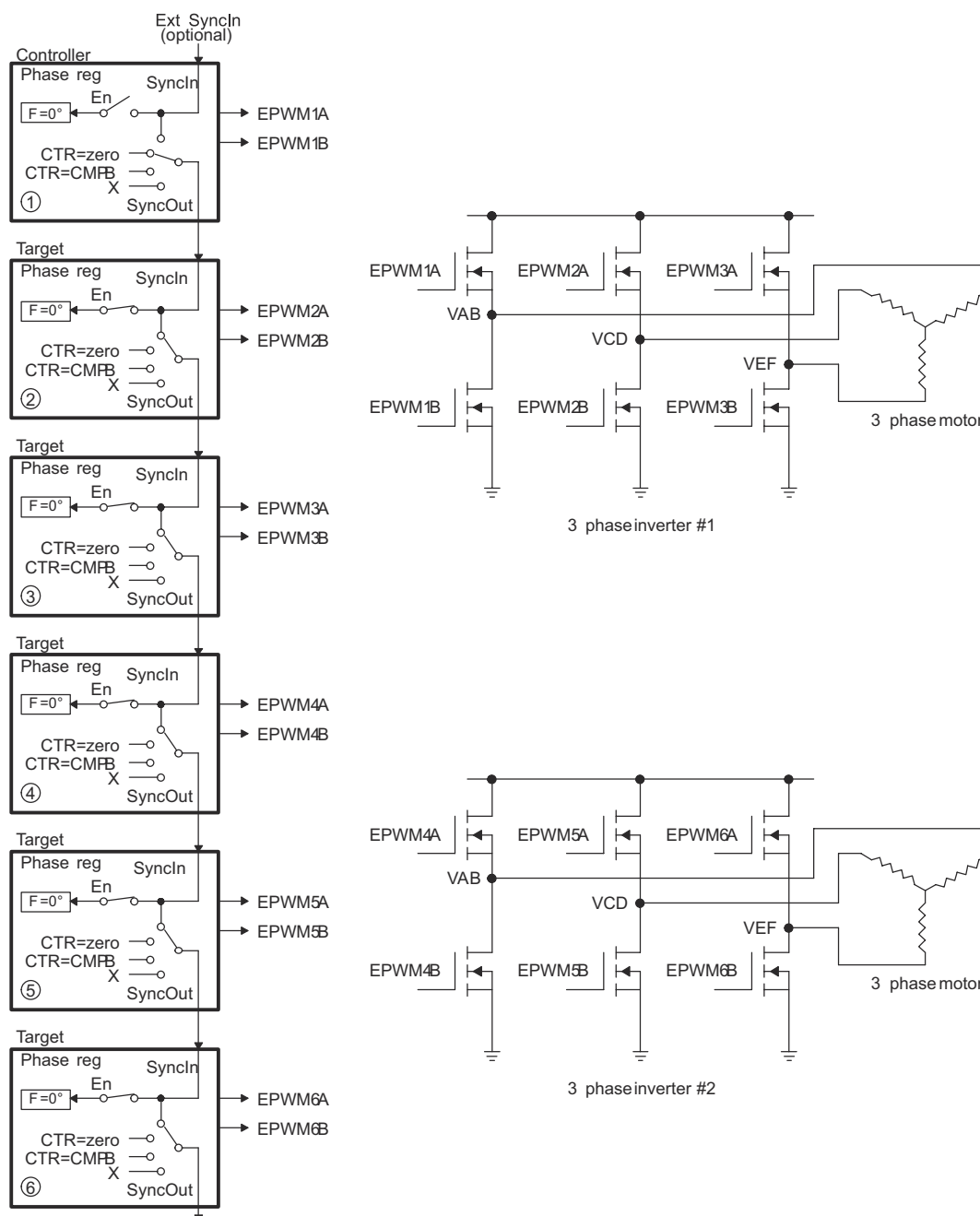
```
//=====
// Config
//=====
// Initialization Time
//=====
// EPWM Module 1 config
EPwm1Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm1Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm1Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm1Regs.TBCTL.bit.PHSEN = TB_DISABLE; // Controller module
EPwm1Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm1Regs.TBCTL.bit.SYNCSEL = TB_CTR_ZERO; // Sync down-stream module
EPwm1Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm1Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm1Regs.AQCTLA.bit.ZRO = AQ_SET; // set actions for EPWM1A
EPwm1Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm1Regs.AQCTLB.bit.ZRO = AQ_CLEAR; // set actions for EPWM1B
EPwm1Regs.AQCTLB.bit.CAD = AQ_SET;
// EPWM Module 2 config
EPwm2Regs.TBPRD = 600; // Period = 1200 TBCLK counts
EPwm2Regs.TBPHS.half.TBPHS = 0; // Set Phase register to zero
EPwm2Regs.TBCTL.bit.CTRMODE = TB_COUNT_UPDOWN; // Symmetrical mode
EPwm2Regs.TBCTL.bit.PHSEN = TB_ENABLE; // Target module
EPwm2Regs.TBCTL.bit.PRDL = TB_SHADOW;
EPwm2Regs.TBCTL.bit.SYNCSEL = TB_SYNC_IN; // sync flow-through
EPwm2Regs.CMPCTL.bit.SHDWAMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.SHDWBMODE = CC_SHADOW;
EPwm2Regs.CMPCTL.bit.LOADAMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.CMPCTL.bit.LOADBMODE = CC_CTR_ZERO; // load on CTR=Zero
EPwm2Regs.AQCTLA.bit.ZRO = AQ_SET; // set actions for EPWM1A
EPwm2Regs.AQCTLA.bit.CAU = AQ_CLEAR;
EPwm2Regs.AQCTLB.bit.ZRO = AQ_CLEAR; // set actions for EPWM1B
EPwm2Regs.AQCTLB.bit.CAD = AQ_SET;
//=====
EPwm1Regs.CMPA.half.CMPA = 400; // adjust duty for output EPWM1A & EPWM1B
EPwm1Regs.CMPB = 200; // adjust point-in-time for ADCSOC trigger
EPwm2Regs.CMPA.half.CMPA = 500; // adjust duty for output EPWM2A & EPWM2B
EPwm2Regs.CMPB = 250; // adjust point-in-time for ADCSOC trigger
```

#### 18.3.5.1

### 18.3.6 Controlling Dual 3-Phase Inverters for Motors (ACI and PMSM)

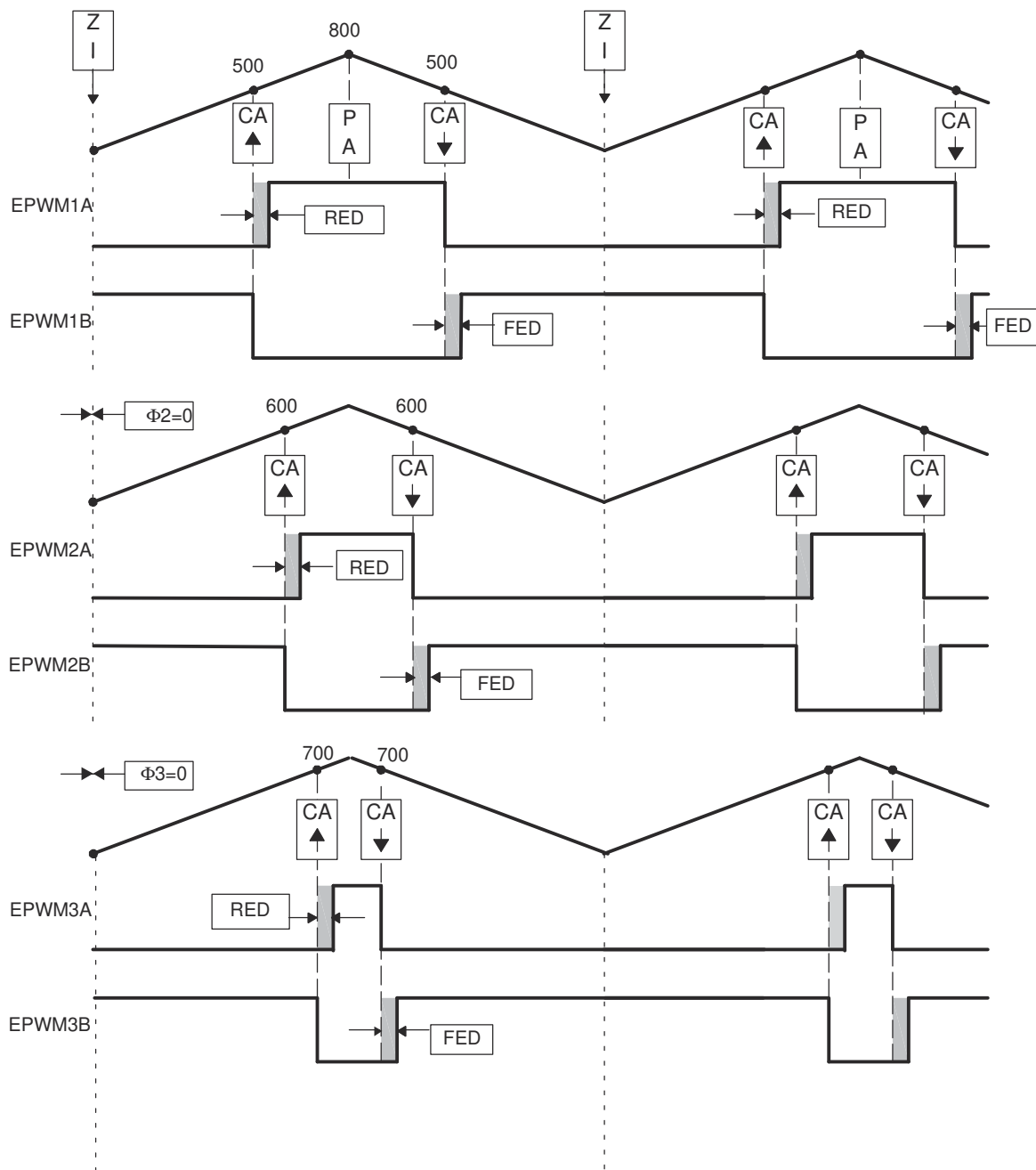
The idea of multiple modules controlling a single power stage can be extended to the 3-phase Inverter case. In such a case, six switching elements can be controlled using three PWM modules, one for each leg of the inverter. Each leg must switch at the same frequency and all legs must be synchronized. A controller + two targets configuration can easily address this requirement. Figure 18-59 shows how six PWM modules can control two independent 3-phase inverters; each running a motor.

As in the cases shown in the previous sections, we have a choice of running each inverter at a different frequency (module 1 and module 4 are controllers as in Figure 18-59), or both inverters can be synchronized by using one controller (module 1) and five targets. In this case, the frequency of modules 4, 5, and 6 (all equal) can be integer multiples of the frequency for modules 1, 2, 3 (also all equal).



**Figure 18-59. Control of Dual 3-Phase Inverter Stages as Is Commonly Used in Motor Control**

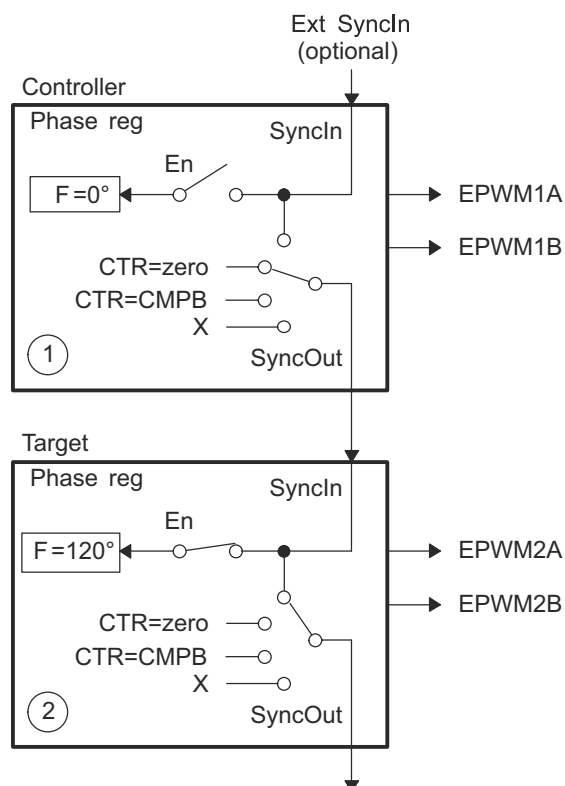




**Figure 18-60. 3-Phase Inverter Waveforms for Figure 18-59 (Only One Inverter Shown)**

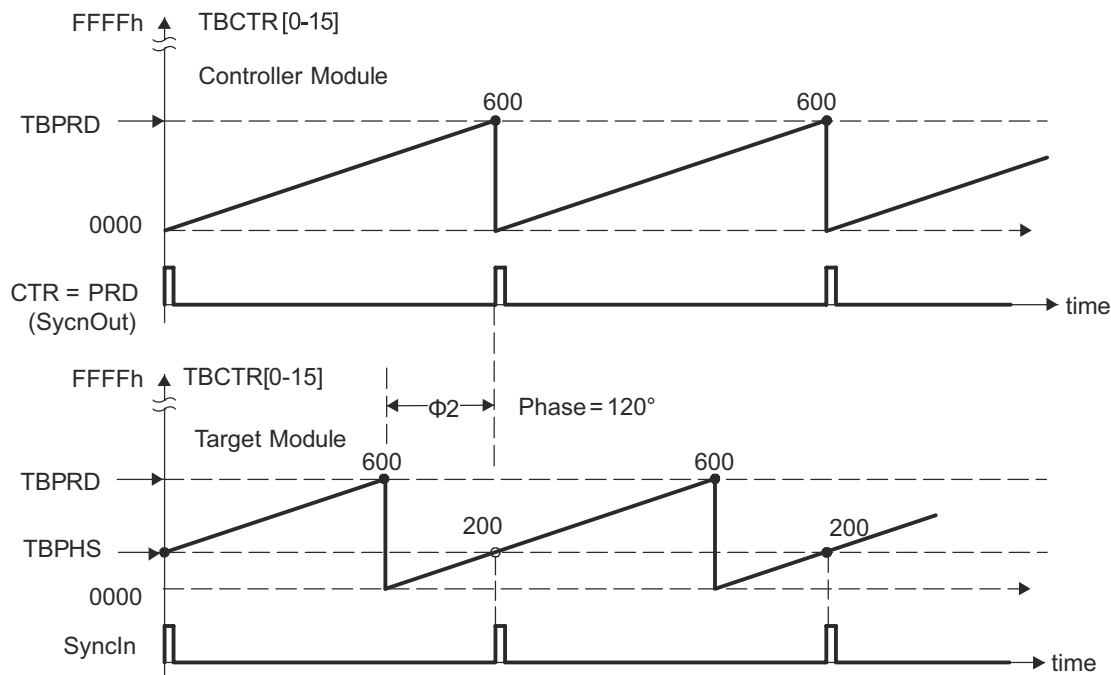
### 18.3.7 Practical Applications Using Phase Control Between PWM Modules

So far, none of the examples have made use of the phase register (TBPHS). It has either been set to zero or its value has been a don't care. However, by programming appropriate values into TBPHS, multiple PWM modules can address another class of applications that rely on phase relationship between stages for correct operation. As described in the TB module section, a PWM module can be configured to allow a SyncIn pulse to cause the TBPHS register to be loaded into the TBCTR register. To illustrate this concept, [Figure 18-61](#) shows a controller and target module with a phase relationship of 120°, that is, the target leads the controller.



**Figure 18-61. Configuring Two PWM Modules for Phase Control**

Figure 18-62 shows the associated timing waveforms for this configuration. Here, TBPRD = 600 for both controller and target. For the target, TBPHS = 200 (that is,  $200/600 \times 360^\circ = 120^\circ$ ). Whenever the controller generates a SyncIn pulse (CTR = PRD), the value of TBPHS = 200 is loaded into the target TBCTR register so the target time-base is always leading the controller's time-base by  $120^\circ$ .



**Figure 18-62. Timing Waveforms Associated With Phase Control Between 2 Modules**

## 18.4 ePWM Module Control and Status Registers

Table 18-22 lists the memory-mapped registers for the ePWM Module Control and Status Registers. All register offset addresses not listed in Table 18-22 should be considered as reserved locations and the register contents should not be modified.

**Table 18-22. ePWM Module Control and Status Registers**

Offset	Acronym	Register Name	Section
0h	TBSTS	Time-Base Status Register	<a href="#">Section 18.4.1</a>
2h	TBCTL	Time-Base Control Register	<a href="#">Section 18.4.2</a>
4h	TBPHS	Time-Base Phase Register	<a href="#">Section 18.4.3</a>
8h	TBPRD	Time-Base Period Register	<a href="#">Section 18.4.4</a>
Ah	TBCTR	Time-Base Counter Register	<a href="#">Section 18.4.5</a>
Ch	CMPCTL	Counter-Compare Control Register	<a href="#">Section 18.4.6</a>
10h	CMPA	Counter-Compare A Register	<a href="#">Section 18.4.7</a>
14h	AQCTLA	Action-Qualifier Control Register for Output A (EPWMxA)	<a href="#">Section 18.4.8</a>
16h	CMPB	Counter-Compare B Register	<a href="#">Section 18.4.9</a>
18h	AQSFRC	Action-Qualifier Software Force Register	<a href="#">Section 18.4.10</a>
1Ah	AQCTLB	Action-Qualifier Control Register for Output B (EPWMxB)	<a href="#">Section 18.4.11</a>
1Ch	DBCTL	Dead-Band Generator Control Register	<a href="#">Section 18.4.12</a>
1Eh	AQCSFRC	Action-Qualifier Continuous S/W Force Register Set	<a href="#">Section 18.4.13</a>
20h	DBFED	Dead-Band Generator Falling Edge Delay Count Register	<a href="#">Section 18.4.14</a>
22h	DBRED	Dead-Band Generator Rising Edge Delay Count Register	<a href="#">Section 18.4.15</a>
24h	TZDCSEL	Trip Zone Digital Compare Event Select Register	<a href="#">Section 18.4.16</a>
26h	TZSEL	Trip-Zone Select Register	<a href="#">Section 18.4.17</a>
28h	TZEINT	Trip-Zone Enable Interrupt Register	<a href="#">Section 18.4.18</a>
2Ah	TZCTL	Trip-Zone Control Register	<a href="#">Section 18.4.19</a>
2Ch	TZCLR	Trip-Zone Clear Register	<a href="#">Section 18.4.20</a>
2Eh	TZFLG	Trip-Zone Flag Register	<a href="#">Section 18.4.21</a>
30h	ETSEL	Event-Trigger Selection Register	<a href="#">Section 18.4.22</a>
32h	TZFRC	Trip-Zone Force Register	<a href="#">Section 18.4.23</a>
34h	ETFLG	Event-Trigger Flag Register	<a href="#">Section 18.4.24</a>
36h	ETPS	Event-Trigger Pre-Scale Register	<a href="#">Section 18.4.25</a>
38h	ETFRC	Event-Trigger Force Register	<a href="#">Section 18.4.26</a>
3Ah	ETCLR	Event-Trigger Clear Register	<a href="#">Section 18.4.27</a>
3Eh	PCCTL	PWM-Chopper Control Register	<a href="#">Section 18.4.28</a>
60h	DCACTL	Digital Compare A Control Register	<a href="#">Section 18.4.29</a>
62h	DCTRIPSEL	Digital Compare Trip Select Register	<a href="#">Section 18.4.30</a>
64h	DCFCTL	Digital Compare Filter Control Register	<a href="#">Section 18.4.31</a>
66h	DCBCTL	Digital Compare B Control Register	<a href="#">Section 18.4.32</a>
68h	DCFOFFSET	Digital Compare Filter Offset Register	<a href="#">Section 18.4.33</a>
6Ah	DCCAPCTL	Digital Compare Capture Control Register	<a href="#">Section 18.4.34</a>
6Ch	DCFWINDOW	Digital Compare Filter Window Register	<a href="#">Section 18.4.35</a>
6Eh	DCFOFFSETCNT	Digital Compare Filter Offset Counter Register	<a href="#">Section 18.4.36</a>
70h	DCCAP	Digital Compare Counter Capture Register	<a href="#">Section 18.4.37</a>
72h	DCFWINDOWCNT	Digital Compare Filter Window Counter Register	<a href="#">Section 18.4.38</a>

### 18.4.1 TBSTS Register (Offset = 0h) [reset = 1h]

TBSTS is shown in [Figure 18-63](#) and described in [Table 18-23](#).

**Figure 18-63. TBSTS Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED					CTRMAX	SYNCl	CTRDlR
R-0h					R-0h	R/W-0h	R-1h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-23. TBSTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-3	RESERVED	R	0h	Reserved
2	CTRMAX	R	0h	Time-Base Counter Max Latched Status Bit 0h = Reading a 0 indicates the time-base counter never reached its maximum value. Writing a 0 will have no effect. 1h = Reading a 1 on this bit indicates that the time-base counter reached the max value 0xFFFF. Writing a 1 to this bit will clear the latched event.
1	SYNCl	R/W	0h	Input Synchronization Latched Status Bit 0h = Writing a 0 will have no effect. Reading a 0 indicates no external synchronization event has occurred. 1h = Reading a 1 on this bit indicates that an external synchronization event has occurred (EPWMxSYNCl). Writing a 1 to this bit will clear the latched event.
0	CTRDlR	R	1h	Time-Base Counter Direction Status Bit. At reset, the counter is frozen, therefore, this bit has no meaning. To make this bit meaningful, you must first set the appropriate mode via TBCTL[CTRMODE]. 0h = Time-Base Counter is currently counting down. 1h = Time-Base Counter is currently counting up.

## 18.4.2 TBCTL Register (Offset = 2h) [reset = 83h]

TBCTL is shown in [Figure 18-64](#) and described in [Table 18-24](#).

**Figure 18-64. TBCTL Register**

15	14	13	12	11	10	9	8
FREE_SOFT		PHSDIR	CLKDIV		HSPCLKDIV		
R/W-0h		R/W-0h	R/W-0h		R/W-1h		
7	6	5	4	3	2	1	0
HSPCLKDIV	SWFSYNC	SYNCOSEL		PRDLD	PHSEN	CTRMODE	
R/W-1h	R/W-0h	R/W-0h		R/W-0h	R/W-0h	R/W-3h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-24. TBCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-14	FREE_SOFT	R/W	0h	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation events: 0h = Stop after the next time-base counter increment or decrement. 1h = Stop when counter completes a whole cycle. In up-count mode, stop when the time-base counter = period (TBCTR = TBPRD). In down-count mode, stop when the time-base counter = 0x0000 (TBCTR = 0x0000). In up-down-count mode, stop when the time-base counter = 0x0000 (TBCTR = 0x0000). 2h = Free run 3h = Free run
13	PHSDIR	R/W	0h	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. The PHSDIR bit indicates the direction the time-base counter (TBCTR) will count after a synchronization event occurs and a new phase value is loaded from the phase (TBPHS) register. This is irrespective of the direction of the counter before the synchronization event. In the up-count and down-count modes this bit is ignored. 0h = Count down after the synchronization event. 1h = Count up after the synchronization event.
12-10	CLKDIV	R/W	0h	Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. $TBCLK = VCLK4 / (HSPCLKDIV \times CLKDIV)$ 0h = /1 (default on reset) 1h = /2 2h = /4 3h = /8 4h = /16 5h = /32 6h = /64 7h = /128

**Table 18-24. TBCTL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9-7	HSPCLKDIV	R/W	1h	High Speed Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. $TBCLK = VCLK4 / (HSPCLKDIV \times CLKDIV)$ 0h = /1 1h = /2 (default on reset) 2h = /4 3h = /6 4h = /8 5h = /10 6h = /12 7h = /14
6	SWFSYNC	R/W	0h	Software Forced Synchronization Pulse. This event is ORed with the EPWMxSYNCl input of the ePWM module. SWFSYNC is valid (operates) only when EPWMxSYNCl is selected by SYNCOSSEL = 0. 0h = Writing a 0 has no effect and reads always return a 0. 1h = Writing a 1 forces a one-time synchronization pulse to be generated.
5-4	SYNCOSSEL	R/W	0h	Synchronization Output Select. These bits select the source of the EPWMxSYNCO signal. 0h = EPWMxSYNCO 1h = CTR = zero: Time-base counter equal to zero (TBCTR = 0x0000) 2h = CTR = CMPB : Time-base counter equal to counter-compare B (TBCTR = CMPB) 3h = Disable EPWMxSYNCO signal
3	PRDL	R/W	0h	Active Period Register Load From Shadow Register Select 0h = The period register (TBPRD) is loaded from its shadow register when the time-base counter, TBCTR, is equal to 0. A write or read to the TBPRD register accesses the shadow register. 1h = Load the TBPRD register immediately without using a shadow register. A write or read to the TBPRD register directly accesses the active register.
2	PHSEN	R/W	0h	Counter Register Load From Phase Register Enable 0h = Do not load the time-base counter (TBCTR) from the time-base phase register (TBPHS) 1h = Load the time-base counter with the phase register when an EPWMxSYNCl input signal occurs or when a software synchronization is forced by the SWFSYNC bit, or when a digital compare sync event occurs.
1-0	CTRM	R/W	3h	Counter Mode. The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 0h = Up-count mode 1h = Down-count mode 2h = Up-down-count mode 3h = Stop-freeze counter operation (default on reset)

### 18.4.3 TBPHS Register (Offset = 4h) [reset = 0h]

TBPHS is shown in [Figure 18-65](#) and described in [Table 18-25](#).

**Figure 18-65. TBPHS Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-25. TBPHS Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	TBPHS	R/W	0h	<p>These bits set time-base counter phase of the selected ePWM relative to the time-base that is supplying the synchronization input signal. Valid values: 0-FFFFh</p> <p>If TBCTL[PHSEN] = 0, then the synchronization event is ignored and the time-base counter is not loaded with the phase.</p> <p>If TBCTL[PHSEN] = 1, then the time-base counter (TBCTR) will be loaded with the phase (TBPHS) when a synchronization event occurs. The synchronization event can be initiated by the input synchronization signal (EPWMxSYNCI) or by a software forced synchronization.</p>



#### 18.4.4 TBPRD Register (Offset = 8h) [reset = 0h]

TBPRD is shown in [Figure 18-66](#) and described in [Table 18-26](#).

**Figure 18-66. TBPRD Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-26. TBPRD Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	TBPRD	R/W	0h	<p>These bits determine the period of the time-base counter. This sets the PWM frequency. Valid values: 0-FFFFh</p> <p>Shadowing of this register is enabled and disabled by the TBCTL[PRDL] bit. By default this register is shadowed.</p> <p>If TBCTL[PRDL] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the time-base counter equals 0.</p> <p>If TBCTL[PRDL] = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware.</p> <p>The active and shadow registers share the same memory map address.</p>

### 18.4.5 TBCTR Register (Offset = Ah) [reset = 0h]

TBCTR is shown in [Figure 18-67](#) and described in [Table 18-27](#).

**Figure 18-67. TBCTR Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBCTR															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-27. TBCTR Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	TBCTR	R/W	0h	Reading these bits gives the current time-base counter value. Valid values: 0-FFFFh Writing to these bits sets the current time-base counter value. The update happens as soon as the write occurs. The write is NOT synchronized to the time-base clock (TBCLK) and the register is not shadowed.

### 18.4.6 CMPCTL Register (Offset = Ch) [reset = 0h]

CMPCTL is shown in [Figure 18-68](#) and described in [Table 18-28](#).

**Figure 18-68. CMPCTL Register**

15	14	13	12	11	10	9	8
RESERVED						SHDWBFULL	SHDWAFULL
R-0h						R-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	
R-0h	R/W-0h	R-0h	R/W-0h	R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-28. CMPCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	SHDWBFULL	R	0h	Counter-compare B (CMPB) Shadow Register Full Status Flag. This bit self clears once a load-strobe occurs. 0h = CMPB shadow FIFO not full yet 1h = Indicates the CMPB shadow FIFO is full a CPU write will overwrite current shadow value.
8	SHDWAFULL	R	0h	Counter-compare A (CMPA) Shadow Register Full Status Flag. The flag bit is set when a 32-bit write to CMPA:CMPAHR register or a 16-bit write to CMPA register is made. A 16-bit write to CMPAHR register will not affect the flag. This bit self clears once a load-strobe occurs. 0h = CMPA shadow FIFO not full yet 1h = Indicates the CMPA shadow FIFO is full, a CPU write will overwrite the current shadow value.
7	RESERVED	R	0h	Reserved
6	SHDWBMODE	R/W	0h	Counter-compare B (CMPB) Register Operating Mode. 0h = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1h = Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action.
5	RESERVED	R	0h	Reserved
4	SHDWAMODE	R/W	0h	Counter-compare A (CMPA) Register Operating Mode. 0h = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 1h = Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action.

**Table 18-28. CMPCTL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
3-2	LOADBMODE	R/W	0h	Active Counter-Compare B (CMPB) Load From Shadow Select Mode. This bit has no effect in immediate mode (CMPCTL[SHDWBMODE] = 1). 0h = Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 1h = Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 2h = Load on either CTR = Zero or CTR = PRD 3h = Freeze (no loads possible)
1-0	LOADAMODE	R/W	0h	Active Counter-Compare A (CMPA) Load From Shadow Select Mode. This bit has no effect in immediate mode (CMPCTL[SHDWAMODE] = 1). 0h = Load on CTR = Zero: Time-base counter equal to zero (TBCTR = 0x0000) 1h = Load on CTR = PRD: Time-base counter equal to period (TBCTR = TBPRD) 2h = Load on either CTR = Zero or CTR = PRD 3h = Freeze (no loads possible)

### 18.4.7 CMPA Register (Offset = 10h) [reset = 0h]

CMPA is shown in [Figure 18-69](#) and described in [Table 18-29](#).

**Figure 18-69. CMPA Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-29. CMPA Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	CMPA	R/W	0h	<p>The value in the active CMPA register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare A" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include:</p> <ul style="list-style-type: none"> <li>Do nothing, the event is ignored.</li> <li>Clear: Pull the EPWMxA and/or EPWMxB signal low</li> <li>Set: Pull the EPWMxA and/or EPWMxB signal high</li> <li>Toggle the EPWMxA and/or EPWMxB signal</li> </ul> <p>Shadowing of this register is enabled and disabled by the CMPCTL[SHDWAMODE] bit. By default this register is shadowed. If CMPCTL[SHDWAMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADAMODE] bit field determines which event will load the active register from the shadow register. Before a write, the CMPCTL[SHDWAFULL] bit can be read to determine if the shadow register is currently full. If CMPCTL[SHDWAMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. In either mode, the active and shadow registers share the same memory map address.</p>

### 18.4.8 AQCTLA Register (Offset = 14h) [reset = 0h]

AQCTLA is shown in [Figure 18-70](#) and described in [Table 18-30](#).

**Figure 18-70. AQCTLA Register**

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-30. AQCTLA Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-10	CBD	R/W	0h	Action when the time-base counter equals the active CMPB register and the counter is decrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxA output low. 2h = Set: force EPWMxA output high. 3h = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.
9-8	CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxA output low. 2h = Set: force EPWMxA output high. 3h = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.
7-6	CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxA output low. 2h = Set: force EPWMxA output high. 3h = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.
5-4	CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxA output low. 2h = Set: force EPWMxA output high. 3h = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.

**Table 18-30. AQCTLA Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	<p>Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down.</p> <p>0h = Do nothing (action disabled).</p> <p>1h = Clear: force EPWMxA output low.</p> <p>2h = Set: force EPWMxA output high.</p> <p>3h = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.</p>
1-0	ZRO	R/W	0h	<p>Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up.</p> <p>0h = Do nothing (action disabled).</p> <p>1h = Clear: force EPWMxA output low.</p> <p>2h = Set: force EPWMxA output high.</p> <p>3h = Toggle EPWMxA output: low output signal will be forced high, and a high signal will be forced low.</p>

### 18.4.9 CMPB Register (Offset = 16h) [reset = 0h]

CMPB is shown in [Figure 18-71](#) and described in [Table 18-31](#).

**Figure 18-71. CMPB Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB															
R/W-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-31. CMPB Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	CMPB	R/W	0h	<p>The value in the active CMPB register is continuously compared to the time-base counter (TBCTR). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare B" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWMxA or the EPWMxB output depending on the configuration of the AQCTLA and AQCTLB registers. The actions that can be defined in the AQCTLA and AQCTLB registers include:</p> <p>Do nothing, event is ignored.</p> <p>Clear: Pull the EPWMxA and/or EPWMxB signal low</p> <p>Set: Pull the EPWMxA and/or EPWMxB signal high</p> <p>Toggle the EPWMxA and/or EPWMxB signal</p> <p>Shadowing of this register is enabled and disabled by the CMPCTL[SHDWBMODE] bit. By default this register is shadowed. If CMPCTL[SHDWBMODE] = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the CMPCTL[LOADBMODE] bit field determines which event will load the active register from the shadow register.</p> <p>Before a write, the CMPCTL[SHDWBFULL] bit can be read to determine if the shadow register is currently full.</p> <p>If CMPCTL[SHDWBMODE] = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware.</p> <p>In either mode, the active and shadow registers share the same memory map address.</p>



#### 18.4.10 AQSFR Register (Offset = 18h) [reset = 0h]

AQSFR is shown in [Figure 18-72](#) and described in [Table 18-32](#).

**Figure 18-72. AQSFR Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RLDCSF		OTSFB	ACTSFB		OTSFA	ACTSFA	
R/W-0h		R/W-0h	R/W-0h		R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-32. AQSFR Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-6	RLDCSF	R/W	0h	AQCSFRC Active Register Reload From Shadow Options 0h = Load on event counter equals zero 1h = Load on event counter equals period 2h = Load on event counter equals zero or counter equals period 3h = Load immediately (the active register is directly accessed by the CPU and is not loaded from the shadow register).
5	OTSFB	R/W	0h	One-Time Software Forced Event on Output B 0h = Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete, that is, a forced event is initiated.) This is a one-shot forced event. It can be overridden by another subsequent event on output B. 1h = Initiates a single s/w forced event
4-3	ACTSFB	R/W	0h	Action when One-Time Software Force B Is Invoked. 0h = Does nothing (action disabled) 1h = Clear (low) 2h = Set (high) 3h = Toggle (Low to High, High to Low). Note: This action is not qualified by counter direction (CNT_dir).
2	OTSFA	R/W	0h	One-Time Software Forced Event on Output A 0h = Writing a 0 has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (that is, a forced event is initiated). 1h = Initiates a single software forced event
1-0	ACTSFA	R/W	0h	Action When One-Time Software Force A Is Invoked. 0h = Does nothing (action disabled) 1h = Clear (low) 2h = Set (high) 3h = Toggle (Low to High, High to Low). Note: This action is not qualified by counter direction (CNT_dir).

### 18.4.11 AQCTLB Register (Offset = 1Ah) [reset = 0h]

AQCTLB is shown in [Figure 18-73](#) and described in [Table 18-33](#).

**Figure 18-73. AQCTLB Register**

15	14	13	12	11	10	9	8
RESERVED				CBD		CBU	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
CAD		CAU		PRD		ZRO	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-33. AQCTLB Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-10	CBD	R/W	0h	Action when the counter equals the active CMPB register and the counter is decrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxB output low. 2h = Set: force EPWMxB output high. 3h = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
9-8	CBU	R/W	0h	Action when the counter equals the active CMPB register and the counter is incrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxB output low. 2h = Set: force EPWMxB output high. 3h = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
7-6	CAD	R/W	0h	Action when the counter equals the active CMPA register and the counter is decrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxB output low. 2h = Set: force EPWMxB output high. 3h = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.
5-4	CAU	R/W	0h	Action when the counter equals the active CMPA register and the counter is incrementing. 0h = Do nothing (action disabled). 1h = Clear: force EPWMxB output low. 2h = Set: force EPWMxB output high. 3h = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.

**Table 18-33. AQCTLB Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
3-2	PRD	R/W	0h	<p>Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down.</p> <p>0h = Do nothing (action disabled).</p> <p>1h = Clear: force EPWMxB output low.</p> <p>2h = Set: force EPWMxB output high.</p> <p>3h = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.</p>
1-0	ZRO	R/W	0h	<p>Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up.</p> <p>0h = Do nothing (action disabled).</p> <p>1h = Clear: force EPWMxB output low.</p> <p>2h = Set: force EPWMxB output high.</p> <p>3h = Toggle EPWMxB output: low output signal will be forced high, and a high signal will be forced low.</p>

### 18.4.12 DBCTL Register (Offset = 1Ch) [reset = 0h]

DBCTL is shown in [Figure 18-74](#) and described in [Table 18-34](#).

**Figure 18-74. DBCTL Register**

15	14	13	12	11	10	9	8
HALFCYCLE	RESERVED						
R/W-0h	R-0h						
7	6	5	4	3	2	1	0
RESERVED		IN_MODE		POLSEL		OUT_MODE	
R-0h		R/W-0h		R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-34. DBCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15	HALFCYCLE	R/W	0h	Half Cycle Clocking Enable Bit: 0h = Full cycle clocking enabled. The dead-band counters are clocked at the TBCLK rate. 1h = Half cycle clocking enabled. The dead-band counters are clocked at TBCLK x 2.
14-6	RESERVED	R	0h	Reserved
5-4	IN_MODE	R/W	0h	Dead Band Input Mode Control. Bit 5 controls the S5 switch and bit 4 controls the S4 switch shown in . This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms the default is EPWMxA In is the source for both falling and rising-edge delays. 0h = EPWMxA In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 1h = EPWMxB In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxA In (from the action-qualifier) is the source for falling-edge delayed signal. 2h = EPWMxA In (from the action-qualifier) is the source for rising-edge delayed signal. EPWMxB In (from the action-qualifier) is the source for falling-edge delayed signal. 3h = EPWMxB In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal.

**Table 18-34. DBCTL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
3-2	POLSEL	R/W	0h	<p>Polarity Select Control.</p> <p>Bit 3 controls the S3 switch and bit 2 controls the S2 switch shown in .</p> <p>This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule.</p> <p>The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that DBCTL[OUT_MODE] = 1,1 and DBCTL[IN_MODE] = 0,0. Other enhanced modes are also possible, but not regarded as typical usage modes.</p> <p>0h = Active high (AH) mode. Neither EPWMxA nor EPWMxB is inverted (default).</p> <p>1h = Active low complementary (ALC) mode. EPWMxA is inverted.</p> <p>2h = Active high complementary (AHC). EPWMxB is inverted.</p> <p>3h = Active low (AL) mode. Both EPWMxA and EPWMxB are inverted.</p>
1-0	OUT_MODE	R/W	0h	<p>Dead-band Output Mode Control.</p> <p>Bit 1 controls the S1 switch and bit 0 controls the S0 switch shown in .</p> <p>This allows you to selectively enable or bypass the dead-band generation for the falling-edge and rising-edge delay.</p> <p>0h = Dead-band generation is bypassed for both output signals. In this mode, both the EPWMxA and EPWMxB output signals from the action-qualifier are passed directly to the PWM-chopper submodule. In this mode, the POLSEL and IN_MODE bits have no effect.</p> <p>1h = Disable rising-edge delay. The EPWMxA signal from the action-qualifier is passed straight through to the EPWMxA input of the PWM-chopper submodule. The falling-edge delayed signal is seen on output EPWMxB. The input signal for the delay is determined by DBCTL[IN_MODE].</p> <p>2h = The rising-edge delayed signal is seen on output EPWMxA. The input signal for the delay is determined by DBCTL[IN_MODE].</p> <p>Disable falling-edge delay. The EPWMxB signal from the action-qualifier is passed straight through to the EPWMxB input of the PWM-chopper submodule.</p> <p>3h = Dead-band is fully enabled for both rising-edge delay on output EPWMxA and falling-edge delay on output EPWMxB. The input signal for the delay is determined by DBCTL[IN_MODE].</p>

### 18.4.13 AQCSFRC Register (Offset = 1Eh) [reset = 0h]

AQCSFRC is shown in [Figure 18-75](#) and described in [Table 18-35](#).

**Figure 18-75. AQCSFRC Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				CSFB		CSFA	
R-0h				R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-35. AQCSFRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3-2	CSFB	R/W	0h	Continuous Software Force on Output B. In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. To configure shadow mode, use AQSFRC[RLDCSF]. 0h = Forcing disabled, that is, has no effect 1h = Forces a continuous low on output B 2h = Forces a continuous high on output B 3h = Software forcing is disabled and has no effect
1-0	CSFA	R/W	0h	Continuous Software Force on Output A. In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. 0h = Forcing disabled, that is, has no effect 1h = Forces a continuous low on output A 2h = Forces a continuous high on output A 3h = Software forcing is disabled and has no effect

#### 18.4.14 DBFED Register (Offset = 20h) [reset = 0h]

DBFED is shown in [Figure 18-76](#) and described in [Table 18-36](#).

**Figure 18-76. DBFED Register**

15	14	13	12	11	10	9	8
RESERVED						DEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
DEL							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-36. DBFED Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	DEL	R/W	0h	Falling Edge Delay Count. 10-bit counter.

### 18.4.15 DBRED Register (Offset = 22h) [reset = 0h]

DBRED is shown in [Figure 18-77](#) and described in [Table 18-37](#).

**Figure 18-77. DBRED Register**

15	14	13	12	11	10	9	8
RESERVED						DEL	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
DEL							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-37. DBRED Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9-0	DEL	R/W	0h	Rising Edge Delay Count. 10-bit counter.



#### 18.4.16 TZDCSEL Register (Offset = 24h) [reset = 0h]

TZDCSEL is shown in [Figure 18-78](#) and described in [Table 18-38](#).

**Figure 18-78. TZDCSEL Register**

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2			DCBEVT1
R-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
DCBEVT1		DCAEVT2			DCAEVT1		
R/W-0h		R/W-0h			R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-38. TZDCSEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-9	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Selection 0h = Event disabled 1h = DCBH = low, DCBL = don't care 2h = DCBH = high, DCBL = don't care 3h = DCBL = low, DCBH = don't care 4h = DCBL = high, DCBH = don't care 5h = DCBL = high, DCBH = low 6h = Reserved 7h = Reserved
8-6	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Selection 0h = Event disabled 1h = DCBH = low, DCBL = don't care 2h = DCBH = high, DCBL = don't care 3h = DCBL = low, DCBH = don't care 4h = DCBL = high, DCBH = don't care 5h = DCBL = high, DCBH = low 6h = Reserved 7h = Reserved
5-3	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Selection 0h = Event disabled 1h = DCAH = low, DCAL = don't care 2h = DCAH = high, DCAL = don't care 3h = DCAL = low, DCAH = don't care 4h = DCAL = high, DCAH = don't care 5h = DCAL = high, DCAH = low 6h = Reserved 7h = Reserved

**Table 18-38. TZDCSEL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2-0	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Selection 0h = Event disabled 1h = DCAH = low, DCAL = don't care 2h = DCAH = high, DCAL = don't care 3h = DCAL = low, DCAH = don't care 4h = DCAL = high, DCAH = don't care 5h = DCAL = high, DCAH = low 6h = Reserved 7h = Reserved

#### 18.4.17 TZSEL Register (Offset = 26h) [reset = 0h]

TZSEL is shown in [Figure 18-79](#) and described in [Table 18-39](#).

One-Shot (OSHT) Trip-zone enable/disable (bits 15-8). When any of the enabled pins go low, a one-shot trip event occurs for this ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EPWMxB outputs. The one-shot trip condition remains latched until the user clears the condition via the TZCLR register. Cycle-by-Cycle (CBC) Trip-zone enable/disable (bits 7-0). When any of the enabled pins go low, a cycle-by-cycle trip event occurs for this ePWM module. When the event occurs, the action defined in the TZCTL register is taken on the EPWMxA and EPWMxB outputs. A cycle-by-cycle trip condition is automatically cleared when the time-base counter reaches zero.

**Figure 18-79. TZSEL Register**

15	14	13	12	11	10	9	8
DCBEVT1	DCAEVT1	OSHT6	OSHT5	OSHT4	OSHT3	OSHT2	OSHT1
R-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
DCBEVT2	DCAEVT2	CBC6	CBC5	CBC4	CBC3	CBC2	CBC1
R-0h	R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-39. TZSEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15	DCBEVT1	R	0h	Digital Compare Output B Event 1 Select 0h = Disable DCBEVT1 as one-shot-trip source for this ePWM module 1h = Enable DCBEVT1 as one-shot-trip source for this ePWM module
14	DCAEVT1	R	0h	Digital Compare Output A Event 1 Select 0h = Disable DCAEVT1 as one-shot-trip source for this ePWM module 1h = Enable DCAEVT1 as one-shot-trip source for this ePWM module
13	OSHT6	R/W	0h	Trip-zone 6 (/TZ6) Select 0h = Disable /TZ6 as a one-shot trip source for this ePWM module 1h = Enable /TZ6 as a one-shot trip source for this ePWM module
12	OSHT5	R/W	0h	Trip-zone 5 (/TZ5) Select 0h = Disable /TZ5 as a one-shot trip source for this ePWM module 1h = Enable /TZ5 as a one-shot trip source for this ePWM module
11	OSHT4	R/W	0h	Trip-zone 4 (/TZ4) Select 0h = Disable /TZ4 as a one-shot trip source for this ePWM module 1h = Enable /TZ4 as a one-shot trip source for this ePWM module
10	OSHT3	R/W	0h	Trip-zone 3 (/TZ3) Select 0h = Disable /TZ3 as a one-shot trip source for this ePWM module 1h = Enable /TZ3 as a one-shot trip source for this ePWM module
9	OSHT2	R/W	0h	Trip-zone 2 (/TZ2) Select 0h = Disable /TZ2 as a one-shot trip source for this ePWM module 1h = Enable /TZ2 as a one-shot trip source for this ePWM module
8	OSHT1	R/W	0h	Trip-zone 1 (/TZ1) Select 0h = Disable /TZ1 as a one-shot trip source for this ePWM module 1h = Enable /TZ1 as a one-shot trip source for this ePWM module

**Table 18-39. TZSEL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
7	DCBEVT2	R	0h	Digital Compare Output B Event 2 Select 0h = Disable DCBEVT2 as a CBC trip source for this ePWM module 1h = Enable DCBEVT2 as a CBC trip source for this ePWM module
6	DCAEVT2	R	0h	Digital Compare Output A Event 2 Select 0h = Disable DCAEVT2 as a CBC trip source for this ePWM module 1h = Enable DCAEVT2 as a CBC trip source for this ePWM module
5	CBC6	R/W	0h	Trip-zone 6 (/TZ6) Select 0h = Disable /TZ6 as a CBC trip source for this ePWM module 1h = Enable /TZ6 as a CBC trip source for this ePWM module
4	CBC5	R/W	0h	Trip-zone 5 (/TZ5) Select 0h = Disable /TZ5 as a CBC trip source for this ePWM module 1h = Enable /TZ5 as a CBC trip source for this ePWM module
3	CBC4	R/W	0h	Trip-zone 4 (/TZ4) Select 0h = Disable /TZ4 as a CBC trip source for this ePWM module 1h = Enable /TZ4 as a CBC trip source for this ePWM module
2	CBC3	R/W	0h	Trip-zone 3 (/TZ3) Select 0h = Disable /TZ3 as a CBC trip source for this ePWM module 1h = Enable /TZ3 as a CBC trip source for this ePWM module
1	CBC2	R/W	0h	Trip-zone 2 (/TZ2) Select 0h = Disable /TZ2 as a CBC trip source for this ePWM module 1h = Enable /TZ2 as a CBC trip source for this ePWM module
0	CBC1	R/W	0h	Trip-zone 1 (/TZ1) Select 0h = Disable /TZ1 as a CBC trip source for this ePWM module 1h = Enable /TZ1 as a CBC trip source for this ePWM module

### 18.4.18 TZEINT Register (Offset = 28h) [reset = 0h]

TZEINT is shown in [Figure 18-80](#) and described in [Table 18-40](#).

**Figure 18-80. TZEINT Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-40. TZEINT Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R/W	0h	Digital Comparator Output B Event 2 Interrupt Enable 0h = Disabled 1h = Enabled
5	DCBEVT1	R/W	0h	Digital Comparator Output B Event 1 Interrupt Enable 0h = Disabled 1h = Enabled
4	DCAEVT2	R/W	0h	Digital Comparator Output A Event 2 Interrupt Enable 0h = Disabled 1h = Enabled
3	DCAEVT1	R/W	0h	Digital Comparator Output A Event 1 Interrupt Enable 0h = Disabled 1h = Enabled
2	OST	R/W	0h	Trip-zone One-Shot Interrupt Enable 0h = Disable one-shot interrupt generation 1h = Enable Interrupt generation a one-shot trip event will cause a EPWMx_TZINT VIM interrupt.
1	CBC	R/W	0h	Trip-zone Cycle-by-Cycle Interrupt Enable 0h = Disable cycle-by-cycle interrupt generation 1h = Enable interrupt generation a cycle-by-cycle trip event will cause an EPWMx_TZINT VIM interrupt.
0	RESERVED	R	0h	Reserved

### 18.4.19 TZCTL Register (Offset = 2Ah) [reset = 0h]

TZCTL is shown in [Figure 18-81](#) and described in [Table 18-41](#).

**Figure 18-81. TZCTL Register**

15	14	13	12	11	10	9	8
RESERVED				DCBEVT2		DCBEVT1	
R-0h				R/W-0h		R/W-0h	
7	6	5	4	3	2	1	0
DCAEVT2		DCAEVT1		TZB		TZA	
R/W-0h		R/W-0h		R/W-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-41. TZCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-12	RESERVED	R	0h	Reserved
11-10	DCBEVT2	R/W	0h	Digital Compare Output B Event 2 Action On EPWMxB: 0h = High-impedance (EPWMxB = High-impedance state) 1h = Force EPWMxB to a high state 2h = Force EPWMxB to a low state 3h = Do Nothing, trip action is disabled
9-8	DCBEVT1	R/W	0h	Digital Compare Output B Event 1 Action On EPWMxB: 0h = High-impedance (EPWMxB = High-impedance state) 1h = Force EPWMxB to a high state 2h = Force EPWMxB to a low state 3h = Do Nothing, trip action is disabled
7-6	DCAEVT2	R/W	0h	Digital Compare Output A Event 2 Action On EPWMxA: 0h = High-impedance (EPWMxA = High-impedance state) 1h = Force EPWMxA to a high state 2h = Force EPWMxA to a low state 3h = Do Nothing, trip action is disabled
5-4	DCAEVT1	R/W	0h	Digital Compare Output A Event 1 Action On EPWMxA: 0h = High-impedance (EPWMxA = High-impedance state) 1h = Force EPWMxA to a high state 2h = Force EPWMxA to a low state 3h = Do Nothing, trip action is disabled
3-2	TZB	R/W	0h	When a trip event occurs the following action is taken on output EPWMxB. Which trip-zone pins can cause an event is defined in the TZSEL register. 0h = High-impedance (EPWMxB = High-impedance state) 1h = Force EPWMxB to a high state 2h = Force EPWMxB to a low state 3h = Do nothing, no action is taken on EPWMxB
1-0	TZA	R/W	0h	When a trip event occurs the following action is taken on output EPWMxA. Which trip-zone pins can cause an event is defined in the TZSEL register. 0h = High-impedance (EPWMxA = High-impedance state) 1h = Force EPWMxA to a high state 2h = Force EPWMxA to a low state 3h = Do nothing, no action is taken on EPWMxA

#### 18.4.20 TZCLR Register (Offset = 2Ch) [reset = 0h]

TZCLR is shown in [Figure 18-82](#) and described in [Table 18-42](#).

**Figure 18-82. TZCLR Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toC1 = Write 1 to clear bit; -n = value after reset

**Table 18-42. TZCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R/W1C	0h	Clear Flag for Digital Compare Output B Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCBEVT2 event trip condition.
5	DCBEVT1	R/W1C	0h	Clear Flag for Digital Compare Output B Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCBEVT1 event trip condition.
4	DCAEVT2	R/W1C	0h	Clear Flag for Digital Compare Output A Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCAEVT2 event trip condition.
3	DCAEVT1	R/W1C	0h	Clear Flag for Digital Compare Output A Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 clears the DCAEVT1 event trip condition.
2	OST	R/W	0h	Clear Flag for One-Shot Trip (OST) Latch 0h = Has no effect. Always reads back a 0. 1h = Clears this Trip (set) condition.
1	CBC	R/W	0h	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0h = Has no effect. Always reads back a 0. 1h = Clears this Trip (set) condition.
0	INT	R/W	0h	Global Interrupt Clear Flag. NOTE: No further EPWMx_TZINT VIM interrupts will be generated until the flag is cleared. If the TZFLG.INT bit is cleared and any of the other flag bits are set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. 0h = Has no effect. Always reads back a 0. 1h = Clears the trip-interrupt flag for this ePWM module, TZFLG.INT.

### 18.4.21 TZFLG Register (Offset = 2Eh) [reset = 0h]

TZFLG is shown in [Figure 18-83](#) and described in [Table 18-43](#).

**Figure 18-83. TZFLG Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	INT
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-43. TZFLG Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R	0h	Latched Status Flag for Digital Compare Output B Event 2 0h = Indicates no trip event has occurred on DCBEVT2 1h = Indicates a trip event has occurred for the event defined for DCBEVT2
5	DCBEVT1	R	0h	Latched Status Flag for Digital Compare Output B Event 1 0h = Indicates no trip event has occurred on DCBEVT1 1h = Indicates a trip event has occurred for the event defined for DCBEVT1
4	DCAEVT2	R	0h	Latched Status Flag for Digital Compare Output A Event 2 0h = Indicates no trip event has occurred on DCAEVT2 1h = Indicates a trip event has occurred for the event defined for DCAEVT2
3	DCAEVT1	R	0h	Latched Status Flag for Digital Compare Output A Event 1 0h = Indicates no trip event has occurred on DCAEVT1 1h = Indicates a trip event has occurred for the event defined for DCAEVT1
2	OST	R	0h	Latched Status Flag for A One-Shot Trip Event. This bit is cleared by writing the appropriate value to the TZCLR register. 0h = No one-shot trip event has occurred. 1h = Indicates a trip event has occurred on a pin selected as a one-shot trip source.
1	CBC	R	0h	Latched Status Flag for Cycle-By-Cycle Trip Event. This bit is cleared by writing the appropriate value to the TZCLR register. 0h = No cycle-by-cycle trip event has occurred. 1h = Indicates a trip event has occurred on a signal selected as a cycle-by-cycle trip source. The TZFLG.CBC bit will remain set until it is manually cleared by the user. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, then CBC will be immediately set again. The specified condition on the signal is automatically cleared when the ePWM time-base counter reaches zero (TBCTR = 0x0000) if the trip condition is no longer present. The condition on the signal is only cleared when the TBCTR = 0x0000 no matter where in the cycle the CBC flag is cleared.



**Table 18-43. TZFLG Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	INT	R	0h	<p>Latched Trip Interrupt Status Flag. No further EPWMx_TZINT VIM interrupts will be generated until this flag is cleared. If the interrupt flag is cleared when either CBC or OST is set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. This bit is cleared by writing the appropriate value to the TZCLR register.</p> <p>0h = Indicates no interrupt has been generated. 1h = Indicates an EPWMx_TZINT VIM interrupt was generated because of a trip condition.</p>

## 18.4.22 ETSEL Register (Offset = 30h) [reset = 0h]

ETSEL is shown in [Figure 18-84](#) and described in [Table 18-44](#).

**Figure 18-84. ETSEL Register**

15	14	13	12	11	10	9	8
SOCBEN	SOCBSEL			SOCAEN	SOCASEL		
R/W-0h	R/W-0h			R/W-0h	R/W-0h		
7	6	5	4	3	2	1	0
RESERVED				INTEN	INTSEL		
R-0h				R/W-0h	R/W-0h		

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-44. ETSEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15	SOCBEN	R/W	0h	Enable the ADC Start of Conversion B (EPWMxSOCB) Pulse 0h = Disable EPWMxSOCB 1h = Enable EPWMxSOCB pulse
14-12	SOCBSEL	R/W	0h	EPWMxSOCB Selection Options. These bits determine when a EPWMxSOCB pulse will be generated. 0h = Enable DCBEVT1.soc event. 1h = Enable event time-base counter equal to zero. (TBCTR = 0x0000). 2h = Enable event time-base counter equal to period (TBCTR = TBPRD). 3h = Enable event time-base counter equal to zero or period (TBCTR = 0x0000 or TBCTR = TBPRD). This mode is useful in up-down count mode. 4h = Enable event time-base counter equal to CMPA when the timer is incrementing. 5h = Enable event time-base counter equal to CMPA when the timer is decrementing. 6h = Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h = Enable event: time-base counter equal to CMPB when the timer is decrementing.
11	SOCAEN	R/W	0h	Enable the ADC Start of Conversion A (EPWMxSOCA) Pulse 0h = Disable EPWMxSOCA 1h = Enable EPWMxSOCA pulse

**Table 18-44. ETSEL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
10-8	SOCASEL	R/W	0h	EPWMxSOCA Selection Options. These bits determine when a EPWMxSOCA pulse will be generated. 0h = Enable DCAEVT1.soc event. 1h = Enable event time-base counter equal to zero. (TBCTR = 0x0000). 2h = Enable event time-base counter equal to period (TBCTR = TBPRD). 3h = Enable event time-base counter equal to zero or period (TBCTR = 0x0000 or TBCTR = TBPRD). This mode is useful in up-down count mode. 4h = Enable event time-base counter equal to CMPA when the timer is incrementing. 5h = Enable event time-base counter equal to CMPA when the timer is decrementing. 6h = Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h = Enable event: time-base counter equal to CMPB when the timer is decrementing.
7-4	RESERVED	R	0h	Reserved
3	INTEN	R/W	0h	Enable ePWM Interrupt (EPWMx_INT) Generation 0h = Disable EPWMx_INT generation 1h = Enable EPWMx_INT generation
2-0	INTSEL	R/W	0h	ePWM Interrupt (EPWMx_INT) Selection Options 0h = Reserved 1h = Enable event time-base counter equal to zero. (TBCTR = 0x0000). 2h = Enable event time-base counter equal to period (TBCTR = TBPRD). 3h = Enable event time-base counter equal to zero or period (TBCTR = 0x0000 or TBCTR = TBPRD). This mode is useful in up-down count mode. 4h = Enable event time-base counter equal to CMPA when the timer is incrementing. 5h = Enable event time-base counter equal to CMPA when the timer is decrementing. 6h = Enable event: time-base counter equal to CMPB when the timer is incrementing. 7h = Enable event: time-base counter equal to CMPB when the timer is decrementing.

### 18.4.23 TZFRC Register (Offset = 32h) [reset = 0h]

TZFRC is shown in [Figure 18-85](#) and described in [Table 18-45](#).

**Figure 18-85. TZFRC Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED	DCBEVT2	DCBEVT1	DCAEVT2	DCAEVT1	OST	CBC	RESERVED
R-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-45. TZFRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-7	RESERVED	R	0h	Reserved
6	DCBEVT2	R/W	0h	Force Flag for Digital Compare Output B Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCBEVT2 event trip condition and sets the TZFLG[DCBEVT2] bit.
5	DCBEVT1	R/W	0h	Force Flag for Digital Compare Output B Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCBEVT1 event trip condition and sets the TZFLG[DCBEVT1] bit.
4	DCAEVT2	R/W	0h	Force Flag for Digital Compare Output A Event 2 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCAEVT2 event trip condition and sets the TZFLG[DCAEVT2] bit.
3	DCAEVT1	R/W	0h	Force Flag for Digital Compare Output A Event 1 0h = Writing 0 has no effect. This bit always reads back 0. 1h = Writing 1 forces the DCAEVT1 event trip condition and sets the TZFLG[DCAEVT1] bit.
2	OST	R/W	0h	Force a One-Shot Trip Event via Software 0h = Writing of 0 is ignored. Always reads back a 0. 1h = Forces a one-shot trip event and sets the TZFLG[OST] bit.
1	CBC	R/W	0h	Force a Cycle-by-Cycle Trip Event via Software 0h = Writing of 0 is ignored. Always reads back a 0. 1h = Forces a cycle-by-cycle trip event and sets the TZFLG[CBC] bit.
0	RESERVED	R	0h	Reserved

#### 18.4.24 ETFLG Register (Offset = 34h) [reset = 0h]

ETFLG is shown in [Figure 18-86](#) and described in [Table 18-46](#).

**Figure 18-86. ETFLG Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0h				R-0h	R-0h	R-0h	R-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-46. ETFLG Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	SOCB	R	0h	Latched ePWM ADC Start-of-Conversion B (EPWMxSOCB) Status Flag 0h = Indicates no EPWMxSOCB event occurred. 1h = Indicates that a start of conversion pulse was generated on EPWMxSOCB. The EPWMxSOCB output will continue to be generated even if the flag bit is set.
2	SOCA	R	0h	Latched ePWM ADC Start-of-Conversion A (EPWMxSOCA) Status Flag. Unlike the ETFLG[INT] flag, the EPWMxSOCA output will continue to pulse even if the flag bit is set. 0h = Indicates no event occurred. 1h = Indicates that a start of conversion pulse was generated on EPWMxSOCA. The EPWMxSOCA output will continue to be generated even if the flag bit is set.
1	RESERVED	R	0h	Reserved
0	INT	R	0h	Latched ePWM Interrupt (EPWMx_INT) Status Flag 0h = Indicates no event occurred. 1h = Indicates that an ePWMx interrupt (EPWMx_INT) was generated. No further interrupts will be generated until the flag bit is cleared. Up to one interrupt can be pending while the ETFLG[INT] bit is still set. If an interrupt is pending, it will not be generated until after the ETFLG[INT] bit is cleared. Refer to Event-Trigger Interrupt Generator figure.

### 18.4.25 ETPS Register (Offset = 36h) [reset = 0h]

ETPS is shown in [Figure 18-87](#) and described in [Table 18-47](#).

**Figure 18-87. ETPS Register**

15	14	13	12	11	10	9	8
SOCBCNT		SOCBPRD		SOCACNT		SOCAPRD	
R-0h		R/W-0h		R-0h		R/W-0h	
7	6	5	4	3	2	1	0
RESERVED				INTCNT		INTPRD	
R-0h				R-0h		R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-47. ETPS Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-14	SOCBCNT	R	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Counter Register. These bits indicate how many selected ETSSEL[SOCBSEL] events have occurred: 0h = No events have occurred. 1h = 1 event has occurred. 2h = 2 events have occurred. 3h = 3 events have occurred.
13-12	SOCBPRD	R/W	0h	ePWM ADC Start-of-Conversion B Event (EPWMxSOCB) Period Select. These bits determine how many selected ETSSEL[SOCBSEL] events need to occur before an EPWMxSOCB pulse is generated. To be generated, the pulse must be enabled (ETSSEL[SOCBEN] = 1). The SOCB pulse will be generated even if the status flag is set from a previous start of conversion (ETFLG[SOCB] = 1). Once the SOCB pulse is generated, the ETPS[SOCBCNT] bits will automatically be cleared. 0h = Disable the SOCB event counter. No EPWMxSOCB pulse will be generated 1h = Generate the EPWMxSOCB pulse on the first event: ETPS[SOCBCNT] = 0,1 2h = Generate the EPWMxSOCB pulse on the second event: ETPS[SOCBCNT] = 1,0 3h = Generate the EPWMxSOCB pulse on the third event: ETPS[SOCBCNT] = 1,1
11-10	SOCACNT	R	0h	ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Counter Register. These bits indicate how many selected ETSSEL[SOCASEL] events have occurred: 0h = No events have occurred. 1h = 1 event has occurred. 2h = 2 events have occurred. 3h = 3 events have occurred.

**Table 18-47. ETPS Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9-8	SOCAPRD	R/W	0h	<p>ePWM ADC Start-of-Conversion A Event (EPWMxSOCA) Period Select. These bits determine how many selected ETSEL[SOCASEL] events need to occur before an EPWMxSOCA pulse is generated. To be generated, the pulse must be enabled (ETSEL[SOCASEN] = 1). The SOCA pulse will be generated even if the status flag is set from a previous start of conversion (ETFLG[SOCA] = 1). Once the SOCA pulse is generated, the ETPS[SOCACNT] bits will automatically be cleared.</p> <p>0h = Disable the SOCA event counter. No EPWMxSOCA pulse will be generated</p> <p>1h = Generate the EPWMxSOCA pulse on the first event: ETPS[SOCACNT] = 0,1</p> <p>2h = Generate the EPWMxSOCA pulse on the second event: ETPS[SOCACNT] = 1,0</p> <p>3h = Generate the EPWMxSOCA pulse on the third event: ETPS[SOCACNT] = 1,1</p>
7-4	RESERVED	R	0h	Reserved
3-2	INTCNT	R	0h	<p>ePWM Interrupt Event (EPWMx_INT) Counter Register. These bits indicate how many selected ETSEL[INTSEL] events have occurred. These bits are automatically cleared when an interrupt pulse is generated. If interrupts are disabled, ETSEL[INT] = 0 or the interrupt flag is set, ETFLG[INT] = 1, the counter will stop counting events when it reaches the period value ETPS[INTCNT] = ETPS[INTPRD].</p> <p>0h = No events have occurred.</p> <p>1h = 1 event has occurred.</p> <p>2h = 2 events have occurred.</p> <p>3h = 3 events have occurred.</p>
1-0	INTPRD	R/W	0h	<p>ePWM Interrupt (EPWMx_INT) Period Select. These bits determine how many selected ETSEL[INTSEL] events need to occur before an interrupt is generated. To be generated, the interrupt must be enabled (ETSEL[INT] = 1). If the interrupt status flag is set from a previous interrupt (ETFLG[INT] = 1) then no interrupt will be generated until the flag is cleared via the ETCLR[INT] bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the ETPS[INTCNT] bits will automatically be cleared.</p> <p>Writing a INTPRD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear.</p> <p>Writing a INTPRD value that is less than the current counter value will result in an undefined state.</p> <p>If a counter event occurs at the same instant as a new zero or non-zero INTPRD value is written, the counter is incremented.</p> <p>0h = Disable the interrupt event counter. No interrupt will be generated and ETFRC[INT] is ignored.</p> <p>1h = Generate an interrupt on the first event INTCNT = 01 (first event)</p> <p>2h = Generate interrupt on ETPS[INTCNT] = 1,0 (second event)</p> <p>3h = Generate interrupt on ETPS[INTCNT] = 1,1 (third event)</p>

### 18.4.26 ETFRC Register (Offset = 38h) [reset = 0h]

ETFRC is shown in [Figure 18-88](#) and described in [Table 18-48](#).

**Figure 18-88. ETFRC Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-48. ETFRC Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	SOCB	R/W	0h	SOCB Force Bit. The SOCB pulse will only be generated if the event is enabled in the ETSEL register. The ETFLG[SOCB] flag bit will be set regardless. 0h = Has no effect. Always reads back a 0. 1h = Generates a pulse on EPWMxSOCB and sets the SOCBFLG bit. This bit is used for test purposes.
2	SOCA	R/W	0h	SOCA Force Bit. The SOCA pulse will only be generated if the event is enabled in the ETSEL register. The ETFLG[SOCA] flag bit will be set regardless. 0h = Writing 0 to this bit will be ignored. Always reads back a 0. 1h = Generates a pulse on EPWMxSOCA and set the SOCAFLG bit. This bit is used for test purposes.
1	RESERVED	R	0h	Reserved
0	INT	R/W	0h	INT Force Bit. The interrupt will only be generated if the event is enabled in the ETSEL register. The INT flag bit will be set regardless. 0h = Writing 0 to this bit will be ignored. Always reads back a 0. 1h = Generates an interrupt on /EPWMxINT and set the INT flag bit. This bit is used for test purposes.



### 18.4.27 ETCLR Register (Offset = 3Ah) [reset = 0h]

ETCLR is shown in [Figure 18-89](#) and described in [Table 18-49](#).

**Figure 18-89. ETCLR Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED				SOCB	SOCA	RESERVED	INT
R-0h				R/W-0h	R/W-0h	R-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-49. ETCLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-4	RESERVED	R	0h	Reserved
3	SOCB	R/W	0h	ePWM ADC Start-of-Conversion B (EPWMxSOCB) Flag Clear Bit 0h = Writing a 0 has no effect. Always reads back a 0. 1h = Clears the ETFLG[SOCB] flag bit.
2	SOCA	R/W	0h	ePWM ADC Start-of-Conversion A (EPWMxSOCA) Flag Clear Bit 0h = Writing a 0 has no effect. Always reads back a 0. 1h = Clears the ETFLG[SOCA] flag bit.
1	RESERVED	R	0h	Reserved
0	INT	R/W	0h	ePWM Interrupt (EPWMx_INT) Flag Clear Bit 0h = Writing a 0 has no effect. Always reads back a 0. 1h = Clears the ETFLG[INT] flag bit and enable further interrupts pulses to be generated.

### 18.4.28 PCCTL Register (Offset = 3Eh) [reset = 0h]

PCCTL is shown in [Figure 18-90](#) and described in [Table 18-50](#).

**Figure 18-90. PCCTL Register**

15	14	13	12	11	10	9	8
RESERVED						CHPDUTY	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
CHPFREQ			OSHTWTH			CHPEN	
R/W-0h			R/W-0h			R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-50. PCCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-11	RESERVED	R	0h	Reserved
10-8	CHPDUTY	R/W	0h	Chopping Clock Duty Cycle 0h = Duty = 1/8 (12.5%) 1h = Duty = 2/8 (25.0%) 2h = Duty = 3/8 (37.5%) 3h = Duty = 4/8 (50.0%) 4h = Duty = 5/8 (62.5%) 5h = Duty = 6/8 (75.0%) 6h = Duty = 7/8 (87.5%) 7h = Reserved
7-5	CHPFREQ	R/W	0h	Chopping Clock Frequency 0h = Divide by 1 (no prescale, = 12.5 MHz at 100 MHz VCLK4) 1h = Divide by 2 (6.25 MHz at 100 MHz VCLK4) 2h = Divide by 3 (4.16 MHz at 100 MHz VCLK4) 3h = Divide by 4 (3.12 MHz at 100 MHz VCLK4) 4h = Divide by 5 (2.50 MHz at 100 MHz VCLK4) 5h = Divide by 6 (2.08 MHz at 100 MHz VCLK4) 6h = Divide by 7 (1.78 MHz at 100 MHz VCLK4) 7h = Divide by 8 (1.56 MHz at 100 MHz VCLK4)
4-1	OSHTWTH	R/W	0h	One-Shot Pulse Width 0h = 1 x VCLK4 / 8 wide ( = 80 nS at 100 MHz VCLK4) 1h = 2 x VCLK4 / 8 wide ( = 160 nS at 100 MHz VCLK4) 2h = 3 x VCLK4 / 8 wide ( = 240 nS at 100 MHz VCLK4) 3h = 4 x VCLK4 / 8 wide ( = 320 nS at 100 MHz VCLK4) 4h = 5 x VCLK4 / 8 wide ( = 400 nS at 100 MHz VCLK4) 5h = 6 x VCLK4 / 8 wide ( = 480 nS at 100 MHz VCLK4) 6h = 7 x VCLK4 / 8 wide ( = 560 nS at 100 MHz VCLK4) 7h = 8 x VCLK4 / 8 wide ( = 640 nS at 100 MHz VCLK4) 8h = 9 x VCLK4 / 8 wide ( = 720 nS at 100 MHz VCLK4) 9h = 10 x VCLK4 / 8 wide ( = 800 nS at 100 MHz VCLK4) Ah = 11 x VCLK4 / 8 wide ( = 880 nS at 100 MHz VCLK4) Bh = 12 x VCLK4 / 8 wide ( = 960 nS at 100 MHz VCLK4) Ch = 13 x VCLK4 / 8 wide ( = 1040 nS at 100 MHz VCLK4) Dh = 14 x VCLK4 / 8 wide ( = 1120 nS at 100 MHz VCLK4) Eh = 15 x VCLK4 / 8 wide ( = 1200 nS at 100 MHz VCLK4) Fh = 16 x VCLK4 / 8 wide ( = 1280 nS at 100 MHz VCLK4)

**Table 18-50. PCCTL Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	CHPEN	R/W	0h	PWM-chopping Enable 0h = Disable (bypass) PWM chopping function 1h = Enable chopping function

### 18.4.29 DCACTL Register (Offset = 60h) [reset = 0h]

DCACTL is shown in [Figure 18-91](#) and described in [Table 18-51](#).

**Figure 18-91. DCACTL Register**

15	14	13	12	11	10	9	8
RESERVED						EVT2FRC_SYNCSEL	EVT2SRCSEL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED				EVT1SYNCE	EVT1SOCE	EVT1FRC_SYNCSEL	EVT1SRCSEL
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-51. DCACTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	EVT2FRC_SYNCSEL	R/W	0h	DCAEVT2 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
8	EVT2SRCSEL	R/W	0h	DCAEVT2 Source Signal Select 0h = Source Is DCAEVT2 Signal 1h = Source Is DCEVTFILT Signal
7-4	RESERVED	R	0h	Reserved
3	EVT1SYNCE	R/W	0h	DCAEVT1 SYNC, Enable/Disable 0h = SYNC Generation Disabled 1h = SYNC Generation Enabled
2	EVT1SOCE	R/W	0h	DCAEVT1 SOC, Enable/Disable 0h = SOC Generation Disabled 1h = SOC Generation Enabled
1	EVT1FRC_SYNCSEL	R/W	0h	DCAEVT1 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
0	EVT1SRCSEL	R/W	0h	DCAEVT1 Source Signal Select 0h = Source Is DCAEVT1 Signal 1h = Source Is DCEVTFILT Signal

### 18.4.30 DCTRIPSEL Register (Offset = 62h) [reset = 0h]

DCTRIPSEL is shown in [Figure 18-92](#) and described in [Table 18-52](#).

**Figure 18-92. DCTRIPSEL Register**

15	14	13	12	11	10	9	8
DCBLCOMPSEL				DCBHCOMPSEL			
R/W-0h				R/W-0h			
7	6	5	4	3	2	1	0
DCALCOMPSEL				DCAHCOMPSEL			
R/W-0h				R/W-0h			

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-52. DCTRIPSEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-12	DCBLCOMPSEL	R/W	0h	Digital Compare B Low Input Select. Defines the source for the DCBL input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. Values not shown are reserved. If a device does not have a particular comparator, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input
11-8	DCBHCOMPSEL	R/W	0h	Digital Compare B High Input Select. Defines the source for the DCBH input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. Values not shown are reserved. If a device does not have a particular comparator, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input
7-4	DCALCOMPSEL	R/W	0h	Digital Compare A Low Input Select. Defines the source for the DCAL input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. Values not shown are reserved. If a device does not have a particular comparator, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input
3-0	DCAHCOMPSEL	R/W	0h	Digital Compare A High Input Select. Defines the source for the DCAH input. The TZ signals, when used as trip signals, are treated as normal inputs and can be defined as active high or active low. Values not shown are reserved. If a device does not have a particular comparator, then that option is reserved. 0h = /TZ1 input 1h = /TZ2 input 2h = /TZ3 input

### 18.4.31 DCFCTL Register (Offset = 64h) [reset = 0h]

DCFCTL is shown in [Figure 18-93](#) and described in [Table 18-53](#).

**Figure 18-93. DCFCTL Register**

15	14	13	12	11	10	9	8
RESERVED				RESERVED			
R-0h				R-0h			
7	6	5	4	3	2	1	0
RESERVED	RESERVED	PULSESEL		BLANKINV	BLANKE	SRCSEL	
R-0h	R-0h	R/W-0h		R/W-0h	R/W-0h	R/W-0h	

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-53. DCFCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-13	RESERVED	R	0h	Reserved
12-8	RESERVED	R	0h	Reserved for TI Test
7	RESERVED	R	0h	Reserved
6	RESERVED	R	0h	Reserved for TI Test
5-4	PULSESEL	R/W	0h	Pulse Select For Blanking and Capture Alignment 0h = Time-base counter equal to period (TBCTR = TBPRD) 1h = Time-base counter equal to zero (TBCTR = 0x0000) 2h = Reserved 3h = Reserved
3	BLANKINV	R/W	0h	Blanking Window Inversion 0h = Blanking window not inverted 1h = Blanking window inverted
2	BLANKE	R/W	0h	Blanking Window Enable/Disable 0h = Blanking window is disabled 1h = Blanking window is enabled
1-0	SRCSEL	R/W	0h	Filter Block Signal Source Select 0h = Source Is DCAEVT1 Signal 1h = Source Is DCAEVT2 Signal 2h = Source Is DCBEVT1 Signal 3h = Source Is DCBEVT2 Signal

### 18.4.32 DCBCTL Register (Offset = 66h) [reset = 0h]

DCBCTL is shown in [Figure 18-94](#) and described in [Table 18-54](#).

**Figure 18-94. DCBCTL Register**

15	14	13	12	11	10	9	8
RESERVED						EVT2FRC_SYNCSEL	EVT2SRCSEL
R-0h						R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
RESERVED				EVT1SYNCE	EVT1SOCE	EVT1FRC_SYNCSEL	EVT1SRCSEL
R-0h				R/W-0h	R/W-0h	R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-54. DCBCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-10	RESERVED	R	0h	Reserved
9	EVT2FRC_SYNCSEL	R/W	0h	DCBEVT2 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
8	EVT2SRCSEL	R/W	0h	DCBEVT2 Source Signal Select 0h = Source Is DCBEVT2 Signal 1h = Source Is DCEVTFILT Signal
7-4	RESERVED	R	0h	Reserved
3	EVT1SYNCE	R/W	0h	DCBEVT1 SYNC, Enable/Disable 0h = SYNC Generation Disabled 1h = SYNC Generation Enabled
2	EVT1SOCE	R/W	0h	DCBEVT1 SOC, Enable/Disable 0h = SOC Generation Disabled 1h = SOC Generation Enabled
1	EVT1FRC_SYNCSEL	R/W	0h	DCBEVT1 Force Synchronization Signal Select 0h = Source Is Synchronous Signal 1h = Source Is Asynchronous Signal
0	EVT1SRCSEL	R/W	0h	DCBEVT1 Source Signal Select 0h = Source Is DCBEVT1 Signal 1h = Source Is DCEVTFILT Signal

### 18.4.33 DCFOFFSET Register (Offset = 68h) [reset = 0h]

DCFOFFSET is shown in [Figure 18-95](#) and described in [Table 18-55](#).

**Figure 18-95. DCFOFFSET Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OFFSET															
R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-55. DCFOFFSET Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	OFFSET	R	0h	Blanking Window Offset. Valid values: 0-FFFFh These 16-bits specify the number of TBCLK cycles from the blanking window reference to the point when the blanking window is applied. The blanking window reference is either period or zero as defined by the DCFCTL[PULSESEL] bit. This offset register is shadowed and the active register is loaded at the reference point defined by DCFCTL[PULSESEL]. The offset counter is also initialized and begins to count down when the active register is loaded. When the counter expires, the blanking window is applied. If the blanking window is currently active, then the blanking window counter is restarted.



### 18.4.34 DCCAPCTL Register (Offset = 6Ah) [reset = 0h]

DCCAPCTL is shown in [Figure 18-96](#) and described in [Table 18-56](#).

**Figure 18-96. DCCAPCTL Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED						SHDWMODE	CAPE
R-0h						R/W-0h	R/W-0h

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-56. DCCAPCTL Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-2	RESERVED	R	0h	Reserved
1	SHDWMODE	R/W	0h	TBCTR Counter Capture Shadow Select Mode 0h = Enable shadow mode. The DCCAP active register is copied to shadow register on a TBCTR = TBPRD or TBCTR = zero event as defined by the DCFCTL[PULSESEL] bit. CPU reads of the DCCAP register will return the shadow register contents. 1h = Active Mode. In this mode the shadow register is disabled. CPU reads from the DCCAP register will always return the active register contents.
0	CAPE	R/W	0h	TBCTR Counter Capture Enable/Disable 0h = Disable the time-base counter capture. 1h = Enable the time-base counter capture.

### 18.4.35 DCFWINDOW Register (Offset = 6Ch) [reset = 0h]

DCFWINDOW is shown in [Figure 18-97](#) and described in [Table 18-57](#).

**Figure 18-97. DCFWINDOW Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WINDOW							
R/W-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-57. DCFWINDOW Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Reserved
7-0	WINDOW	R/W	0h	Blanking Window Width. Valid values: 0-FFh specifies the width of the blanking window in TBCLK cycles. The blanking window begins when the offset counter expires. When this occurs, the window counter is loaded and begins to count down. If the blanking window is currently active and the offset counter expires, the blanking window counter is restarted. The blanking window can cross a PWM period boundary. 0h = No blanking window is generated.

### 18.4.36 DCFFSETCNT Register (Offset = 6Eh) [reset = 0h]

DCFFSETCNT is shown in [Figure 18-98](#) and described in [Table 18-58](#).

**Figure 18-98. DCFFSETCNT Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FFSETCNT															
R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-58. DCFFSETCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	FFSETCNT	R	0h	Blanking Offset Counter. Valid values: 0-FFFFh These 16-bits are read only and indicate the current value of the offset counter. The counter counts down to zero and then stops until it is re-loaded on the next period or zero event as defined by the DCFCTL[PULSESEL] bit. The offset counter is not affected by the free/soft emulation bits. That is, it will always continue to count down if the device is halted by a emulation stop.

### 18.4.37 DCCAP Register (Offset = 70h) [reset = 0h]

DCCAP is shown in [Figure 18-99](#) and described in [Table 18-59](#).

**Figure 18-99. DCCAP Register**

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCCAP															
R-0h															

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-59. DCCAP Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-0	DCCAP	R	0h	<p>Digital Compare Time-Base Counter Capture. Valid values: 0-FFFFh</p> <p>To enable time-base counter capture, set the DCCAPCLT[CAPE] bit to 1.</p> <p>If enabled, reflects the value of the time-base counter (TBCTR) on the low to high edge transition of a filtered (DCEVTFLT) event. Further capture events are ignored until the next period or zero as selected by the DCFCTL[PULSESEL] bit.</p> <p>Shadowing of DCCAP is enabled and disabled by the DCCAPCTL[SHDWMODE] bit. By default this register is shadowed. If DCCAPCTL[SHDWMODE] = 0, then the shadow is enabled. In this mode, the active register is copied to the shadow register on the TBCTR = TBPRD or TBCTR = zero as defined by the DCFCTL[PULSESEL] bit. CPU reads of this register will return the shadow register value.</p> <p>If DCCAPCTL[SHDWMODE] = 1, then the shadow register is disabled. In this mode, CPU reads will return the active register value.</p> <p>The active and shadow registers share the same memory map address.</p>

### 18.4.38 DCFWINDOWCNT Register (Offset = 72h) [reset = 0h]

DCFWINDOWCNT is shown in [Figure 18-100](#) and described in [Table 18-60](#).

**Figure 18-100. DCFWINDOWCNT Register**

15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
WINDOWCNT							
R-0h							

LEGEND: R/W = Read/Write; R = Read only; W1toCl = Write 1 to clear bit; -n = value after reset

**Table 18-60. DCFWINDOWCNT Register Field Descriptions**

Bit	Field	Type	Reset	Description
15-8	RESERVED	R	0h	Any writes to these bit(s) must always have a value of 0.
7-0	WINDOWCNT	R	0h	Blanking Window Counter. Valid value: 0-FFh These 8 bits are read only and indicate the current value of the window counter. The counter counts down to zero and then stops until it is re-loaded when the offset counter reaches zero again.



The debug subsystem contains one MCUDebugss at its core that allows the JTAG interface access to device components. The debug subsystem is designed to provide the following debug features:

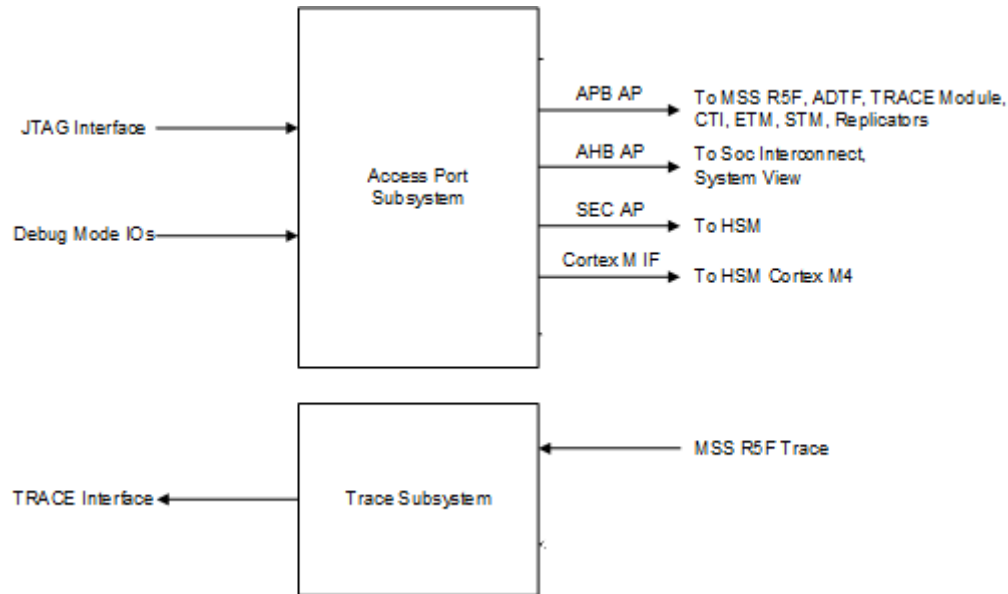
- JTAG debug access to debug resources, mapped through an ARM SWJ-DP and TI ICEPickM scan module
- System memory access without halting the processor
- ETM-based trace for ARM R5F
- Cross trigger to halt and restart MSS based on events such as watchdog, timers, DMA, and time-stamp events
- Capability to read the device ID

<b>19.1 AWR2544 DebugSS Architecture .....</b>	<b>4825</b>
--	-------------

## 19.1 AWR2544 DebugSS Architecture

### 19.1.1 Overview

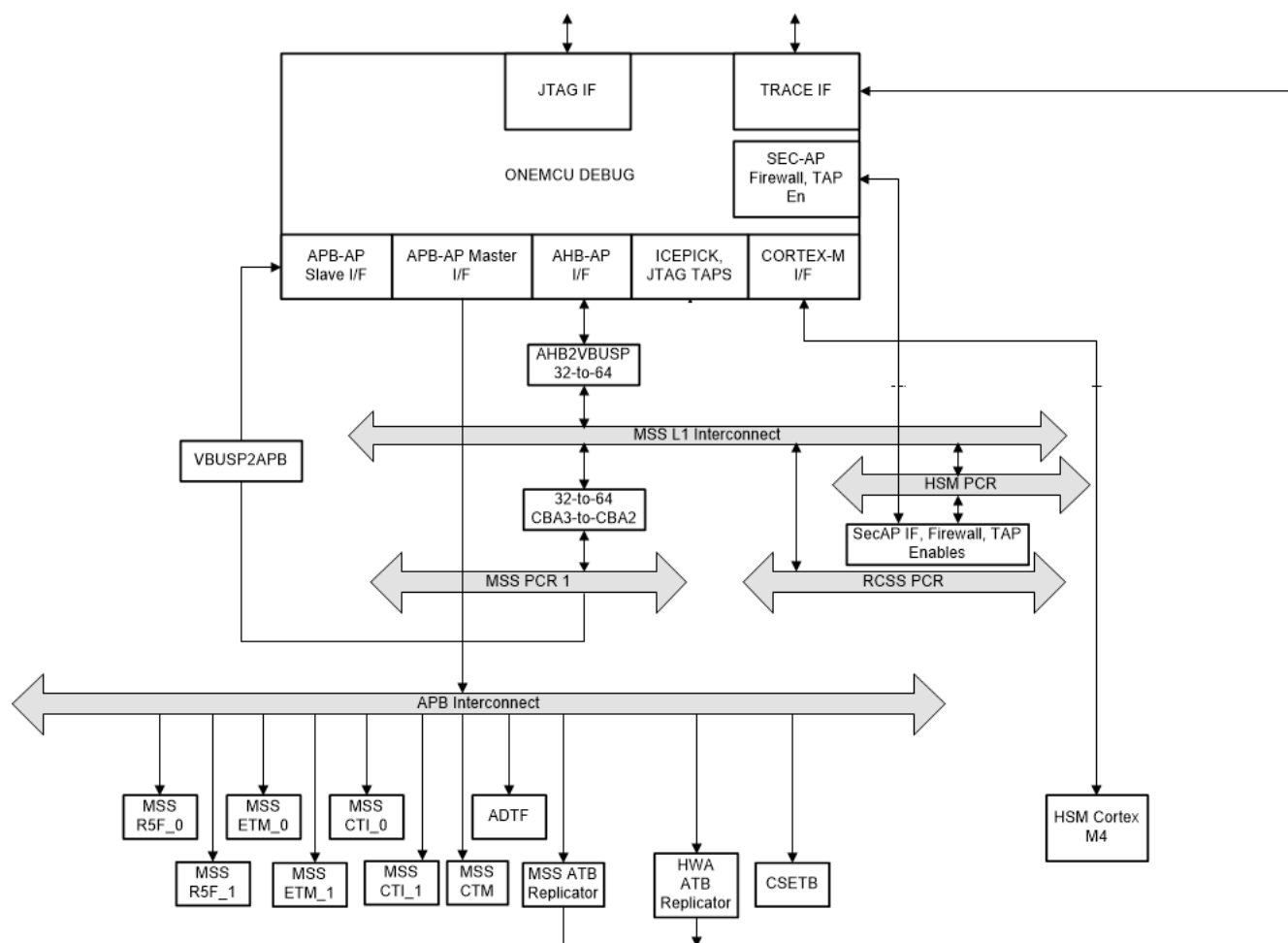
The OneMCU Debug Subsystem (OneMCUDebugSS) is used in the AWR2544 platform. An overview of the interconnectivity of the debug ports and trace ports are shown in [Figure 19-1](#).



**Figure 19-1. Debug SS Overview**

### 19.1.2 DebugSS Architecture

The DebugSS architecture and the connectivity of the various debug components are shown in [Figure 19-2](#).



**Figure 19-2. AWR2544 DebugSSArchitecture**

### 19.1.3 Debug Subsystem Address Map

The memory map view for RS232 and DAP AHB is the same as MSS CortexR5. [Table 19-1](#) shows the APB Port Address Map for the AWR2544.

**Table 19-1. Debug Subsystem Address Map**

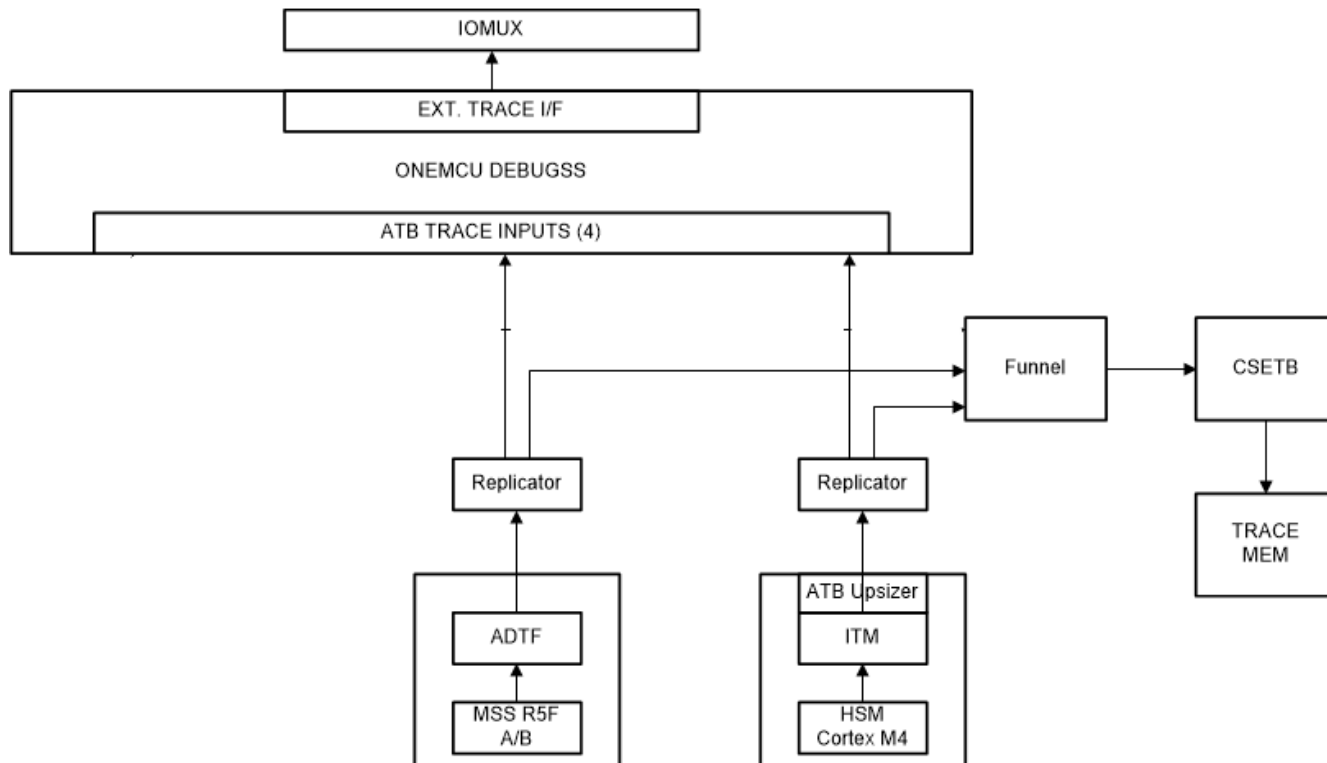
APB Port	Block Name	Start Address Offset	End Address Offset
APB EXTERNAL PORT 0	MSS CR5 ATB REPLICATOR	0x00012000	0x00012FFF
APB EXTERNAL PORT 0	CSETB	0x00013000	0x00013FFF
APB EXTERNAL PORT 0	HSM CM4 REPLICATOR	0x00015000	0x00015FFF
APB EXTERNAL PORT 0	HSM CM4 CTI	0x00018000	0x00018FFF
APB EXTERNAL PORT 1	MSS CR5 ROM Table	0x00020000	0x00020FFF
APB EXTERNAL PORT 2	MSS CR5 C0	0x00030000	0x00030FFF
APB EXTERNAL PORT 2	MSS CR5 C0 CTI	0x00038000	0x00038FFF
APB EXTERNAL PORT 2	MSS CR5 C0 ETM	0x0003C000	0x0003CFFF



## 19.1.4 Trace Subsystem

### 19.1.4.1 Trace Infrastructure

The trace infrastructure is shown in [Figure 19-3](#). Trace data from the MSS CR5 cores and HSM Cortex M4 cores may be routed to the external trace lines over the TPIU-Trace interface. Alternately, there is an 8KB trace memory to store the configured (filtered) trace data onto the memory.



**Figure 19-3. Trace Infrastructure**

The CortexM4 ITM output requires an 8-bit to 32-bit ATB upsizer in the path to convert the signals to be compatible as input to the upstream layers/modules.

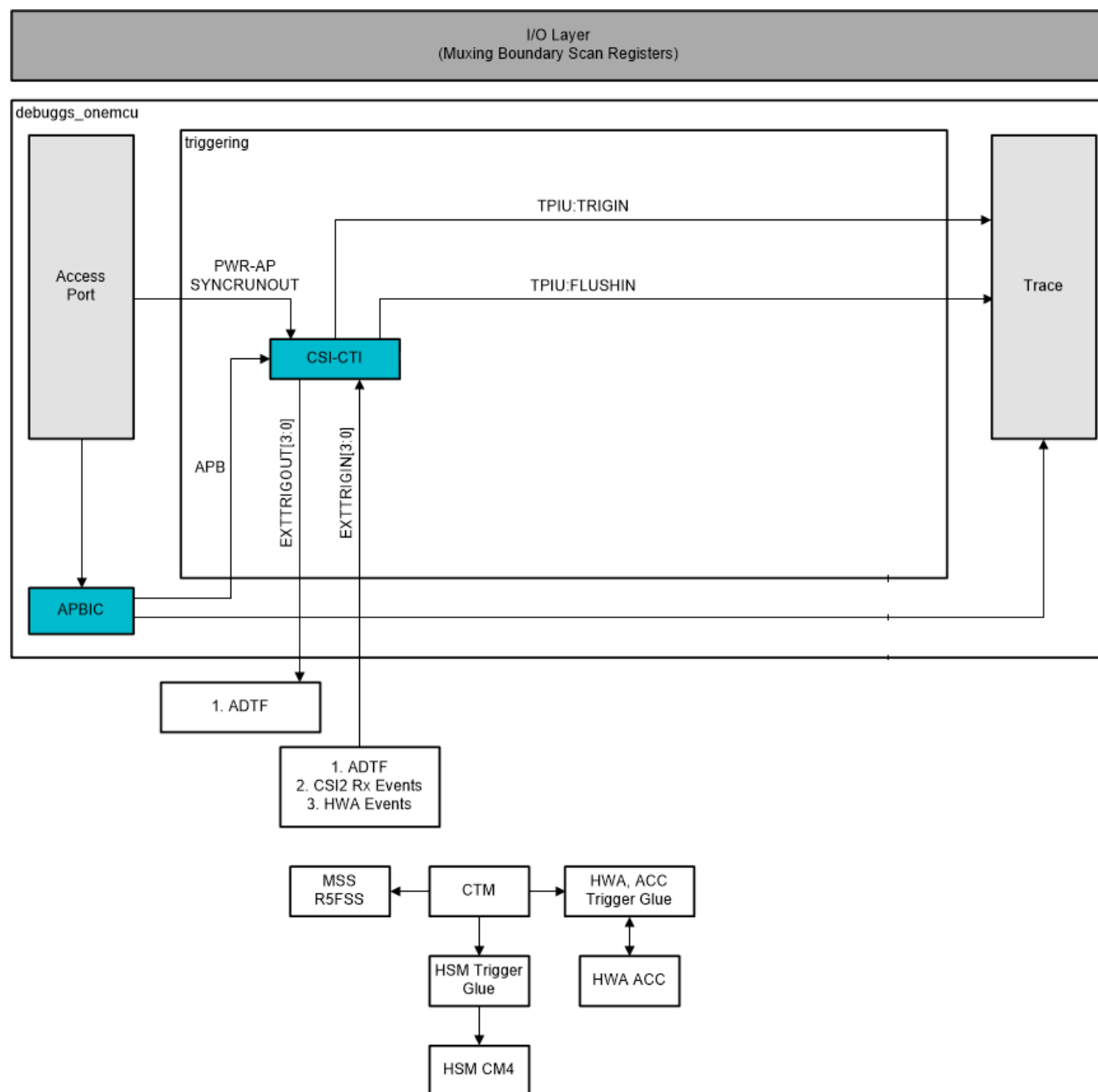
The maximum support frequency for the external interface trace clock is 125 MHz.

### 19.1.4.2 Cross Triggering

#### 19.1.4.2.1 Cross Triggering Infrastructure

The cross triggering infrastructure is shown in [Figure 19-4](#). The host processors involved in the cross triggering sequence are:

- MSS Cortex R5F cores, ETB
- Host as suspend peripherals
  - HWA Accelerator


**Figure 19-4. Cross Triggering Infrastructure**

#### 19.1.4.2.1.1 MSS CR5 CTI Trigger Input Connections

**Table 19-2. MSS CR5 CTI Trigger Input Connections**

CTI Trigger Input Bit	AWR2544 Integration	Description
[7]	MSS_INT_MAP	MSS CR5 Interrupt. Select any of the 256 MSS CR5 Interrupts from Section 7.1. Configure the require interrupt as trigger by writing to MSS_CTRL:: MSS_CTL_TRIG_SEL:MSS_CTL_TRIG_SEL_TRIG8_SEL
[6]	ETMTRIGGER	ETM managed Trigger. Generated internal to Cortex R5 Subsystem
[5]	COMMTX	Communications channel transmit. Generated internal to Cortex R5 Subsystem

**Table 19-2. MSS CR5 CTI Trigger Input Connections (continued)**

CTI Trigger Input Bit	AWR2544 Integration	Description
[4]	COMMRX	Communications channel receive. Generated internal to Cortex R5 Subsystem
[3]	ETMEXTOUT[1]	ETM managed External Output Event 1. Generated internal to Cortex R5 Subsystem
[2]	ETMEXTOUT[0]	ETM managed External Output Event 0. Generated internal to Cortex R5 Subsystem
[1]	PMUIRQ	Interrupt request from performance monitoring unit. Generated internal to Cortex R5 Subsystem
[0]	DBGTRIGGER	CPU is entering the debug state (halted). Generated internal to Cortex R5 Subsystem

#### 19.1.4.2.1.2 MSS CR5 CTI Trigger Output Connections

**Table 19-3. MSS CR5 CTI Trigger Output Connections**

CTI Trigger Output Bit	Integration	Description
[7]	Not Used	
[6]	ETMTRIGACK	ETM
[5]	Not Used	
[4]	DBGRESTART	External restart request
[3]	InIRQ	CPU Interrupt Request
[2]	EXTIN[1]	ETM External Input 1
[1]	EXTIN[0]	ETM External Input 0
[0]	EDBGRQ	External halt request

#### 19.1.4.2.2 OneMCU DebugSS CTI Cross Triggering

##### 19.1.4.2.2.1 OneMCU Debugss CTI Trigger Input Connections

**Table 19-4. OneMCU Debugss CTI Trigger Input Connections**

CTI Trigger Input Bit	AWR2544 Integration	Source
[7]	MSS_CR5_INT_MAP	Select any of the 256 MSS CR5 Interrupts from Section. Configure the require interrupt as trigger by writing to MSS_CTRL::MSS_DEBUGSS_CTI_TRIG_SEL: TRIG3
[6]	MSS_CR5_INT_MAP	Select any of the 256 MSS CR5 Interrupts from Section. Configure the require interrupt as trigger by writing to MSS_CTRL::MSS_DEBUGSS_CTI_TRIG_SEL: TRIG2
[5]	MSS_CR5_INT_MAP	Select any of the 256 MSS CR5 Interrupts from Section. Configure the require interrupt as trigger by writing to MSS_CTRL::MSS_DEBUGSS_CTI_TRIG_SEL: TRIG1
[4]	Reserved	Reserved
[3]	Reserved	ETM. Generated internal to Cortex R5 Subsystem
[2]	Reserved	ETM. Generated internal to Cortex R5 Subsystem
[1]	Reserved	Core. Generated internal to Cortex R5 Subsystem
[0]	PWR-AP:SYNCRUNOUT	Core. Generated internal to Cortex R5 Subsystem

##### 19.1.4.2.2.2 OneMCU Debugss CTI Trigger Output Connections

**Table 19-5. OneMCU Debugss CTI Trigger Output Connections**

CTI Trigger Output Bit	AWR2544 Integration	Destination
[7]	Not Used	
[6]	Not Used	
[5]	Not Used	
[4]	CS-ET:TRIGIN	AWR2544 Embedded Trace Buffer

**Table 19-5. OneMCU Debugss CTI Trigger Output Connections (continued)**

CTI Trigger Output Bit	AWR2544 Integration	Destination
[3]	Reserved	
[2]	Reserved	
[1]	TPIU:FLUSHIN	Internal TPIU
[0]	TPIU:TRIGIN	Internal TPIU

### 19.1.5 Suspend Peripherals

The device supports a suspend feature, which provides a way to stop a "closely coupled" hardware process running on a peripheral-IP when the host processor enters a debug state. The suspend mechanism is important for debug to ensure that peripheral-IPs operate in a lock-step manner with a host controller processor.

**Table 19-6. Suspend Peripherals**

Subsystems Peripherals	CPU Suspend	Peripherals	Peripheral Control Register
MSS Peripherals	MSS CR5 Cores	MSS_CPSW	MSS_CTRL:DBG_ACK_CTL0_CPSW
		MSS_WDT	MSS_CTRL:DBG_ACK_CTL1_WDT
		MSS_RTIA	MSS_CTRL:DBG_ACK_CTL1_RTI
		MSS_RTIB	MSS_CTRL:DBG_ACK_CTL1_RTI
		MSS_RTIC	MSS_CTRL:DBG_ACK_CTL1_RTI
		MSS_MCRC	MSS_CTRL:DBG_ACK_CTL1_MCRC
		MSS_I2C	MSS_CTRL:DBG_ACK_CTL1_I2C
		MSS_SCIA	MSS_CTRL:DBG_ACK_CTL1_SCIA
		MSS_SCIB	MSS_CTRL:DBG_ACK_CTL1_SCIB



20.1 Core Clock Comparator (CCC).....	4832
20.2 Dual Clock Comparator (DCC).....	4833
20.3 ECC_AGGREGATOR.....	4852
20.4 ECC Registers.....	4859
20.5 Error Signaling Module (ESM).....	4964
20.6 Cyclic Redudancy Check (CRC).....	5010
20.7 Self-Test Controller (STC).....	5094
20.8 Programmable Built-In Self-Test (PBIST).....	5178

## 20.1 Core Clock Comparator (CCC)

### 20.1.1 Description

Core Clock Comparator (CCC) supports single-shot and continuous mode of operation, such as DCC. In continuous mode, the programmed values are reloaded after every successful comparison.

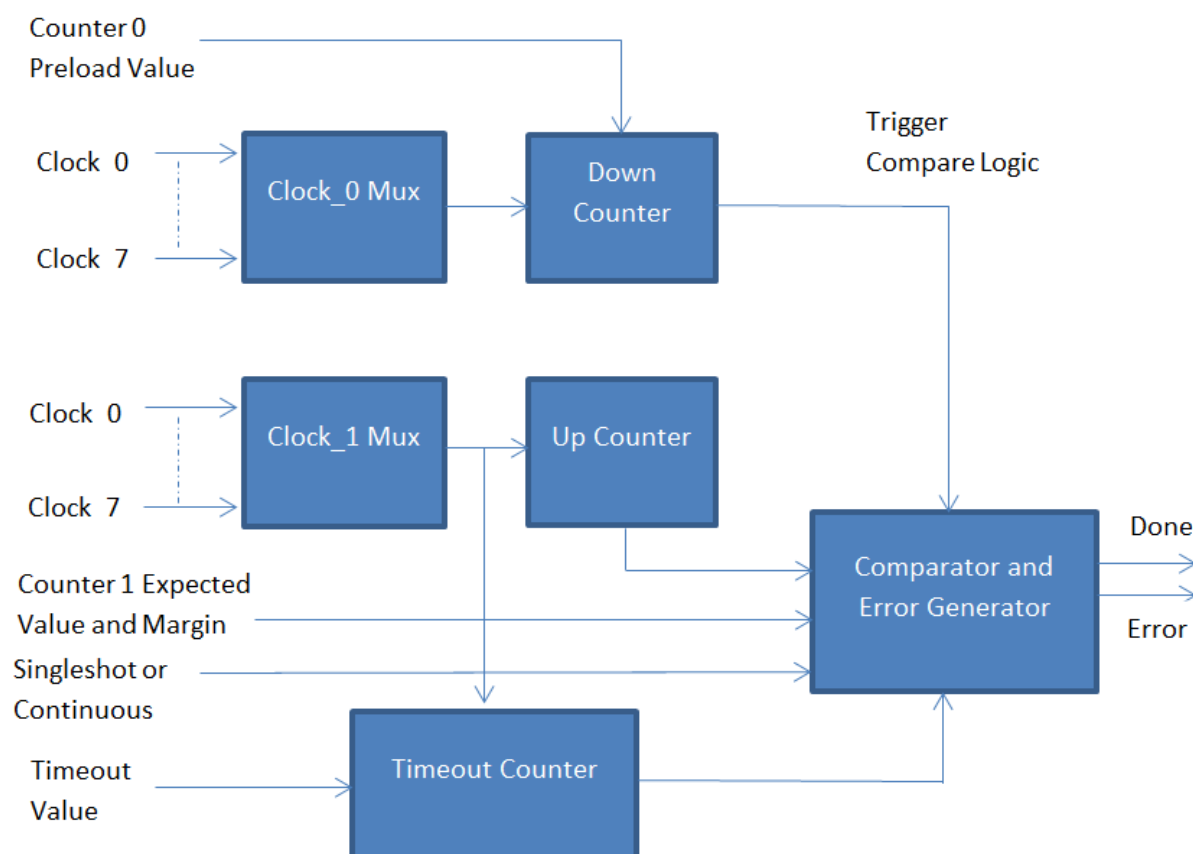
The module accepts 7 clock inputs for clock 0 and clock 1. One of these input clocks is selected to counter 0 and counter 1. Counter 0 is a down counter and is preloaded with a value before enabling the module. Counter 1 is an up counter which operates on Clock 1.

At the expiry of counter 0, value in the counter 1 is compared against the programmed expected value of the counter. After a successful comparison, a Done signal is asserted in single-shot mode, whereas in continuous mode, counter 0 is reloaded for the next comparison. Margin value programmed provides the tolerance for the comparison. An error signal is asserted when the counter 1 value differs from the expected value beyond the tolerance range.

When an error occurs, the module stops comparison in both single-shot and continuous mode.

There is a timeout counter functioning on Clock 1. A timeout value must be loaded into the timeout counter before enabling the module. If the timeout counter expires before the expiry of counter 0, an error condition is indicated. In continuous mode, after the successful comparison, the timeout value is also reloaded along with Counter 0.

### 20.1.2 Block Diagram



**Figure 20-1. CCC Block Diagram**

### 20.1.3 Perform Clock Comparison

The following are the steps to perform clock comparison.

1. Select Clock 0.
2. Select Clock 1.
3. Load value for down counting in counter 0.
4. Load expected value of counter 1.
5. Load Margin value for tolerance.
6. Set singleshot or continuous mode.
7. Load timeout value.
8. Enable Module.
9. Wait for Done or Error indication.

### 20.1.4 I/O Table

**Table 20-1. I/O Table**

Signal	Input/Output	Description
async_rst_n	Input	Module Reset
vbusp_clk	Input	Clock
clock0_src	Input	Input clocks for Counter 0
clock1_src	Input	Input clocks for Counter 1
clock0_sel	Input	Clock Selector for Counter 0
clock1_sel	Input	Clock Selector for Counter 1
count0_expiry_val	Input	Preload Value for Counter 0
count1_expected_val	Input	Expected value of Counter 1
disable_clk_output	Input	Cutoff clocks for the module
enable_module	Input	Enable for Clock comparator
margin_count	Input	Tolerance value for Comparator
singleshot_mode	Input	0->Continuous mode, 1-> Singleshot mode
count1_val_out	Output	Counter 1 value
counter_error	Output	Error Indicator
counter_done	Output	Successful comparison Indicator
mod_status	Output	Internal Status Indicator
async_error_indicator	Output	Error indicator without the synchronizer
timeout_err_count	Input	Preload value for Timeout Counter
atpg_reset_bypass	Input	DFT reset
atpg_clk	Input	DFT clock
icg_te	Input	Control for ICG
atpg_clk_bypass	Input	Control for ATPG clock

### 20.1.5 Recommended Programming

- Clock source 1 must be faster than Clock source 0 for successful comparison of clocks.
- The timeout value must always be loaded for successful comparison. The timeout value must be greater than the duration of the comparison operation.

## 20.2 Dual Clock Comparator (DCC)

This section describes the dual-clock comparator (DCC) module.

## 20.2.1 Introduction

The primary purpose of a DCC module is to measure the frequency of a clock signal using a second known clock signal as a reference. Specifically, DCC is designed to detect drifts from the expected clock frequency. This capability can be used to ensure the correct frequency range for several different device clock sources, thereby enhancing the system safety metrics.

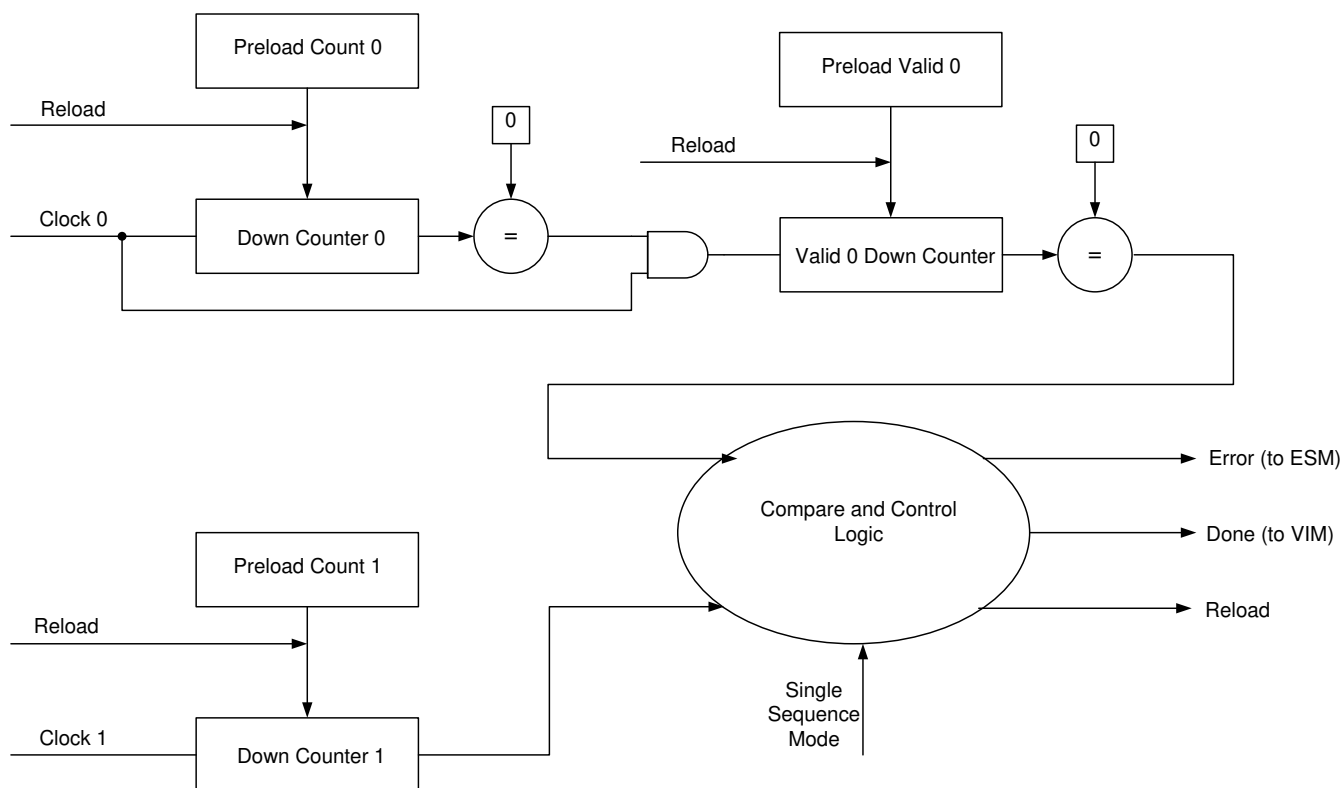
### 20.2.1.1 Main Features

The main features of each of the DCC modules are:

- Allows application to ensure that a fixed ratio is maintained between frequencies of two clock signals
- Supports the definition of a programmable tolerance window in terms of number of reference clock cycles
- Supports continuous monitoring without requiring application intervention
- Also supports a single-sequence mode for spot measurements
- Allows selection of clock source for each of the counters resulting in several specific use cases

### 20.2.1.2 Block Diagram

Figure 20-2 illustrates the main concept of the DCC module.



**Figure 20-2. Block Diagram**



## 20.2.2 Module Operation

As shown in [Figure 20-2](#), the DCC contains two counters – counter0 and counter1, which are driven by two signals – clock0 and clock1. The application programs the seed values for both these counters. The application also configures the tolerance window time by configuring the valid counter for clock0.

Counter0 and counter1 both start counting simultaneously once the DCC is enabled. When counter0 counts down to zero, this automatically triggers the count down of the tolerance window counter (valid0).

The DCC module can be used in two different operating modes:

### 20.2.2.1 Continuous Monitoring Mode

In this mode, the DCC is used by the application to ensure that two clock signals maintain the correct frequency ratio. Suppose the application wants to ensure that the PLL output signal (clock source # 1) always maintains a fixed frequency relationship with the main oscillator (clock source # 0).

- In this case, the application can use the main oscillator as the clock0 signal (for counter0 and valid0) and the PLL output as the clock1 (for counter1).
- The seed values of counter0, valid0 and counter1 are selected such that if the actual frequencies of clock0 and clock1 are equal to their expected frequencies, then the counter1 will reach zero either at the same time as counter0 or during the count down of the valid0 counter.
- If the counter1 reaches zero during the count down of the valid0 counter, then all the counters (counter0, valid0, counter1) are reloaded with their initial seed values once valid0 has also counted down to zero.
- This sequence of counting down and checking then continues as long as there is no error, or until the DCC module is disabled.
- The counters also all get reloaded if the application resets and restarts the DCC module.

#### Error Conditions:

An error condition is generated by any one of the following:

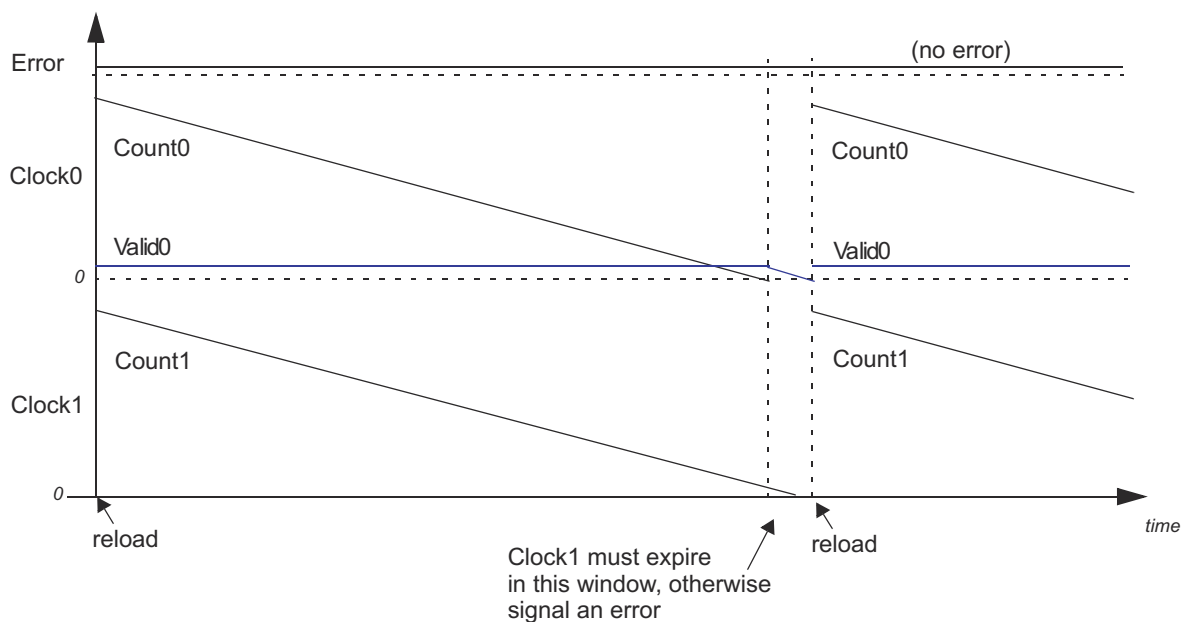
1. Counter1 counts down to 0 before Counter0 reaches 0. This means that clock1 is faster than expected, or clock0 is slower than expected. It includes the case when clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that clock1 is slower than expected. It includes the case when clock1 is stuck at 1 or 0.

Any error freezes the counters from counting. An application may then read out the counter values to help determine what caused the error.

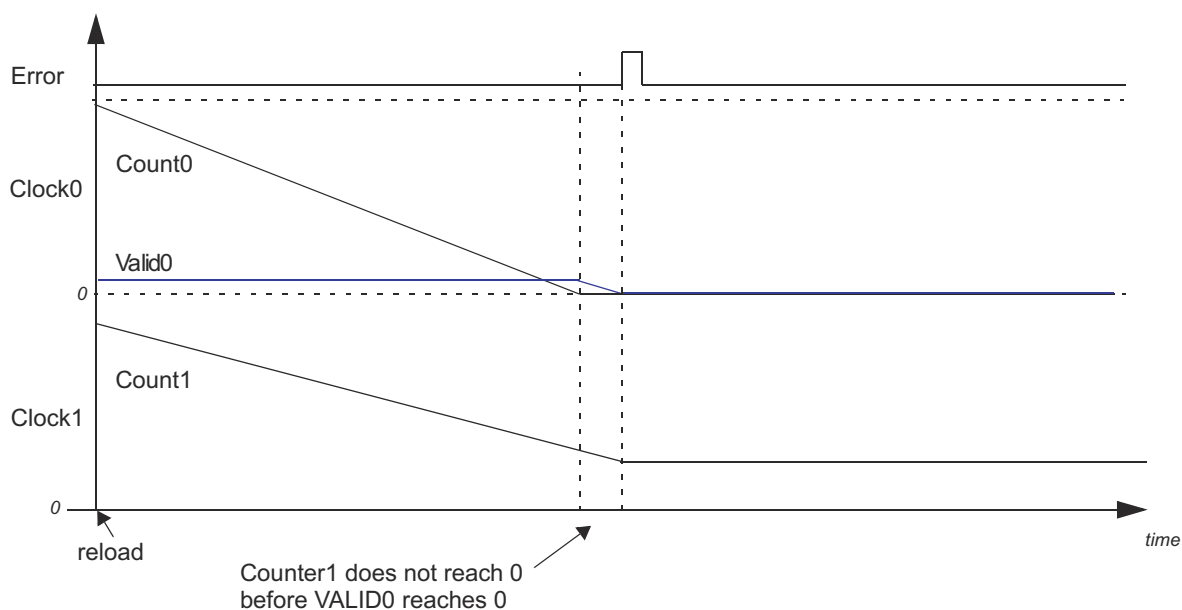
#### 20.2.2.1.1 Error Conditions

While operating in continuous mode, the counters get reloaded with the seed values and continue counting down under the following conditions:

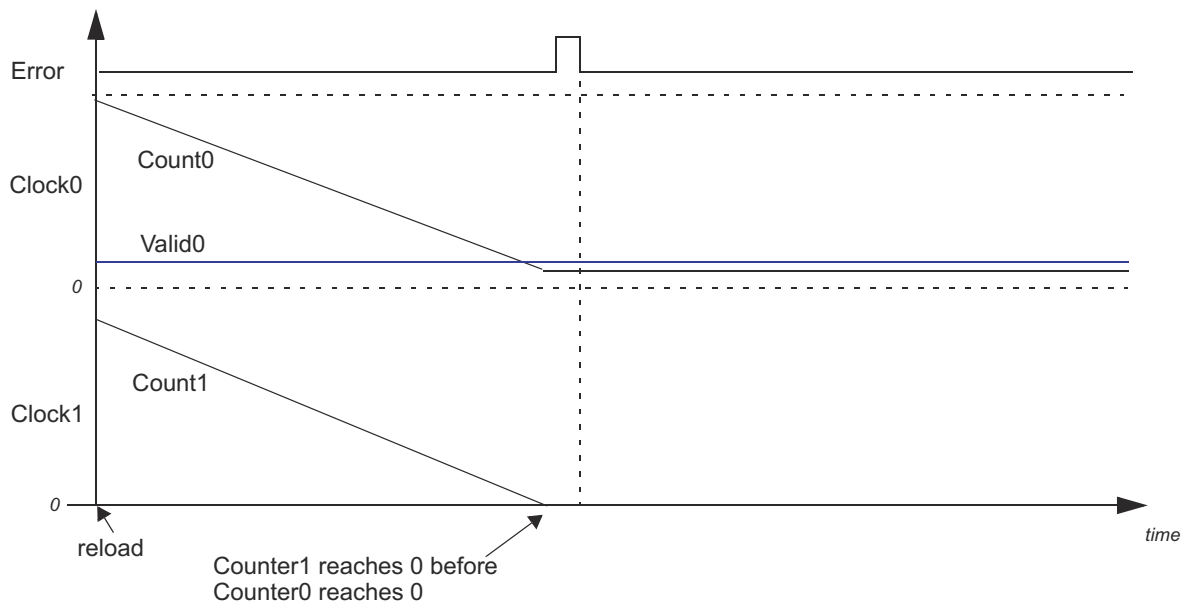
- The module is reset or restarted by the application, OR
- Counter0, Valid 0 and Counter1 all reach 0 without any error



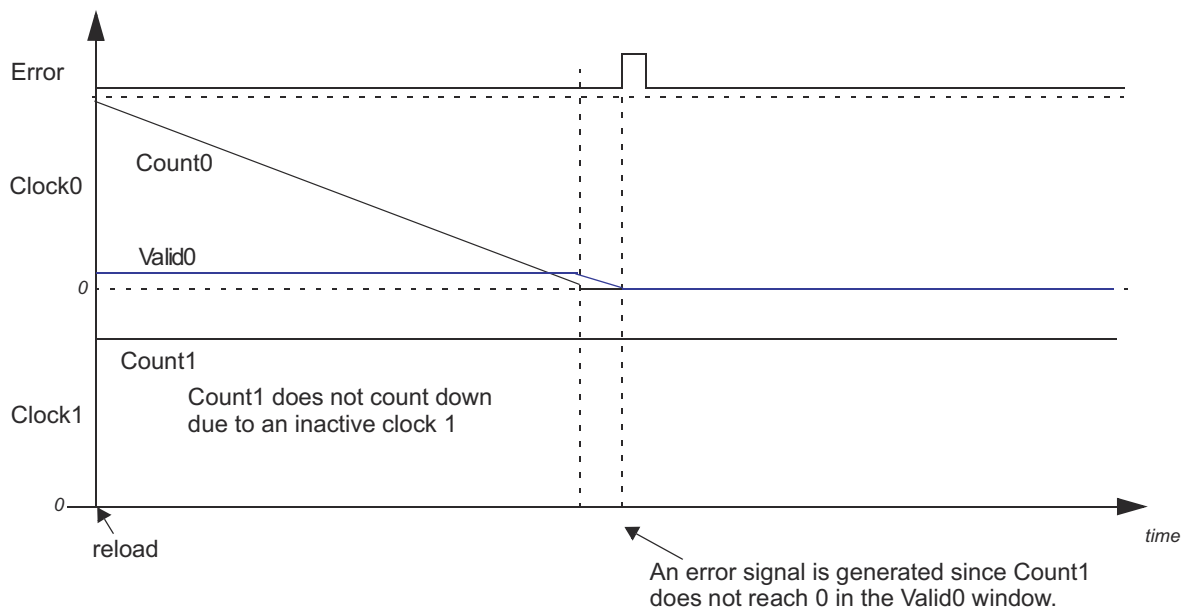
**Figure 20-3. Counter Relationship**



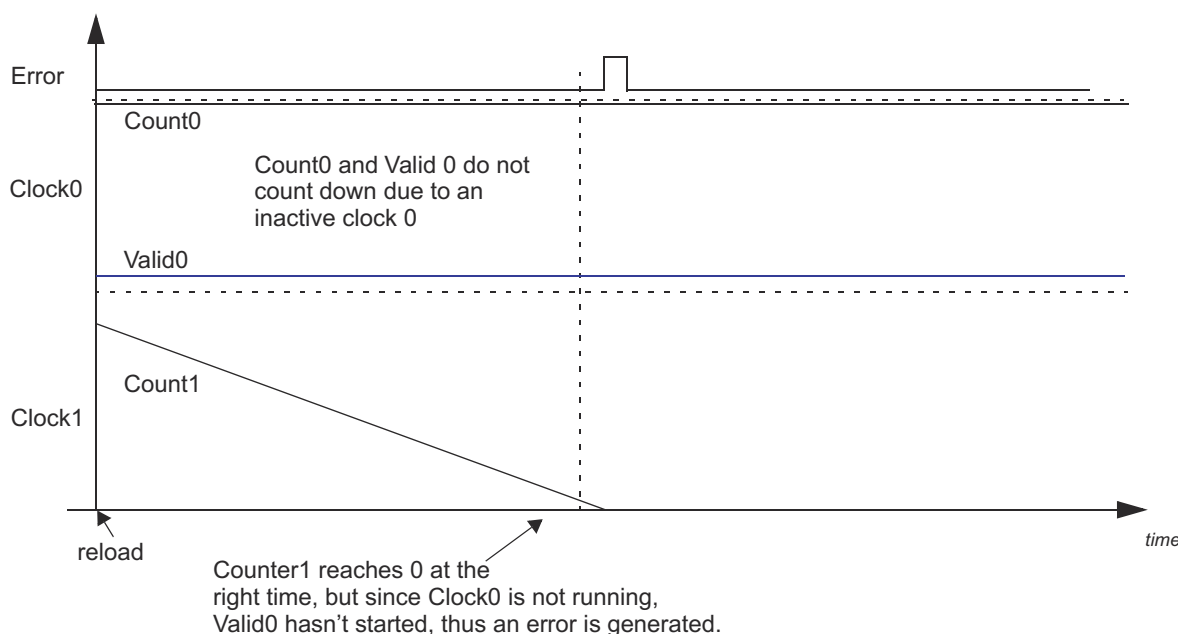
**Figure 20-4. Clock1 Slower Than Clock0 - Results in an Error and Stops Counting**



**Figure 20-5. Clock1 Faster Than Clock0 - Results in an Error and Stops Counting**



**Figure 20-6. Clock1 Not Present - Results in an Error and Stops Counting**



**Figure 20-7. Clock0 Not Present - Results in an Error and Stops Counting**

### 20.2.2.2 Single-Shot Measurement Mode

The DCC module can be programmed to count down one time by enabling the single-shot mode. In this mode, the DCC stops operating when the down counter0 and the valid counter0 reach 0. Alternatively, the DCC can be programmed to stop counting when the down counter1 reaches 0.

At the end of one sequence of counting down in this single-shot mode, the DCC gets disabled automatically, which prevents further counting. This mode is typically used for spot measurements of the frequency of a signal. This frequency could be an unknown for the application before the measurement.

### Example Usage of Single-Shot Measurement Mode: Trimming the High-Frequency Low-Power Oscillator

A practical example of the usage of the spot measurement mode is in trimming the HF LPO (clock source # 5) using the main oscillator as a reference. This measurement sequence would proceed as follows:

- The application sets up the seed values for counter0 and valid0 for the duration of the measurement. Suppose the main oscillator frequency is 10 MHz and the intended duration of the measurement is 500  $\mu$ s. The application needs to configure a seed value of 5000.
- These 5000 counts need to be divided between the counter0 and the valid0 counters. The minimum value for the valid0 seed is 4, so the application can configure counter0 seed value as 4996 and the valid0 seed value as 4.
- Suppose the HF LPO frequency is truly unknown. In this case the application can choose the maximum allowed seed value for counter1. This increases the probability of counter0 and valid0 counting down while the counter1 has still not fully counted down to zero. The maximum allowed seed value for counter1 is 1048575.
- Once the DCC is enabled, the counters counter0 and counter1 both start counting down from their seed values.
- When counter0 reaches zero, it automatically triggers the valid0 counter.
- When valid0 reaches zero, if counter1 is not zero as well, an ERROR status flag is set and a "DCC error" is sent to the ESM. Counter1 is also frozen so that it stops counting down any further. The application can enable an interrupt to be generated from the ESM whenever this DCC error is indicated. Refer the device datasheet to identify the ESM group and channel where the DCC error is connected.
- The DCC error interrupt service routine can then check the value of counter1 when the error was generated. Suppose that the counter1 now reads 1044575. This means that counter1 has counted 1048575 - 1044575,

or 4000 cycles within the 500- $\mu$ s measurement period. This means that the average frequency of the HF LPO over this 500- $\mu$ s period was 4000 cycles / 500  $\mu$ s, or 8 MHz.

- The application then needs to clear the ERROR status flag and restart the DCC module so that it is ready for the next spot measurement.

If there is no error generated at the end of the sequence, then the DONE status flag is set and a DONE interrupt is generated. The application must clear the DONE flag before restarting the DCC.

The conditions that cause a DCC error are identical between the continuous monitoring mode and the single-shot measurement mode.

### Error Conditions:

An error condition is generated by any one of the following:

1. Counter1 counts down to 0 before Counter0 reaches 0. This means that clock1 is faster than expected, or clock0 is slower than expected. It includes the case when clock0 is stuck at 1 or 0.
2. Counter1 does not reach 0 even when Counter0 and Valid0 have both reached 0. This means that clock1 is slower than expected. It includes the case when clock1 is stuck at 1 or 0.

Any error freezes the counters from counting. An application may then read out the counter values to help determine what caused the error.

### Freezing Counters when Counter1 Reaches Zero:

The DCC module also allows the counters to be frozen when the counter1 reaches zero. This allows one of the clock sources for counter1 to be used as a reference for measuring one of the clock sources for counter0. The error conditions are the same as those where (counter0=0 and valid0=0) define the condition when the DCC counters are frozen. That is, an error is indicated if counter0 and valid0 become zero while counter1 is still non-zero. In this case, however, the application would typically set up the seed values such that the counter1 will become zero before counter0. Essentially the measurement period is defined by the seed value of the counter1. Note that this is also an error condition, and the interrupt service routine can use the measurement period and the actual cycles counted by counter1 to determine the frequency of the clock0 signal.

### 20.2.3 MSS DCC Integration

4-DCC modules have been instantiated in the SOC as part of MSS. Clocks to the module are mentioned in the following sections.

#### 20.2.3.1 Input Clock sources

**Table 20-2. MSS\_DCCA Clocking**

Clock Signal	Description	Source	REG Value for Selection
DCC_INPUT00_CLK	Primary Oscillator Clock	XTALCLK	0x0
DCC_INPUT01_CLK	Primary Oscillator Clock	RCCLK10	0x1
DCC_CLKSRC0_CLK	Counter 1 Clock Source	PLL_CORE_HSDIV0_CLKOUT2	0x0
DCC_CLKSRC1_CLK	Counter 1 Clock Source	MSS_CR5F_CLK	0x1
DCC_CLKSRC2_CLK	Counter 1 Clock Source	SYS_CLK	0x2
DCC_CLKSRC3_CLK	Counter 1 Clock Source	MSS_QSPI_CLK	0x3
DCC_CLKSRC4_CLK	Counter 1 Clock Source	MSS_RTIA_CLK	0x4
DCC_CLKSRC7_CLK	Counter 1 Clock Source	MSS_SCIA_CLK	0x7

**Table 20-3. MSS\_DCCB Clocking**

Clock Signal	Description	Source	REG Value for Selection
DCC_INPUT00_CLK	Primary Oscillator Clock	XTALCLK	0x0
DCC_INPUT01_CLK	Primary Oscillator Clock	RCCLK10	0x1
DCC_INPUT02_CLK	Primary Oscillator Clock	SYS_CLK	0x2
DCC_CLKSRC0_CLK	Counter 1 Clock Source	PLL_CORE_HSDIV0_CLKOUT1	0x0

**Table 20-3. MSS\_DCCB Clocking (continued)**

Clock Signal	Description	Source	REG Value for Selection
DCC_CLKSRC1_CLK	Counter 1 Clock Source	MSS_CR5F_CLK	0x1
DCC_CLKSRC2_CLK	Counter 1 Clock Source	XTALCLK	0x2
DCC_CLKSRC3_CLK	Counter 1 Clock Source	MSS_CPTS_CLK	0x3
DCC_CLKSRC4_CLK	Counter 1 Clock Source	MSS_RTIB_CLK	0x4
DCC_CLKSRC5_CLK	Counter 1 Clock Source	MSS_SPIB_CLK	0x5
DCC_CLKSRC6_CLK	Counter 1 Clock Source	MSS_MCANA_CLK	0x6
DCC_CLKSRC7_CLK	Counter 1 Clock Source	MSS_MCANB_CLK	0x7

**Table 20-4. MSS\_DCCC Clocking**

Clock Signal	Description	Source	REG Value for Selection
DCC_INPUT00_CLK	Primary Oscillator Clock	XTALCLK	0x0
DCC_INPUT01_CLK	Primary Oscillator Clock	RCCLK10	0x1
DCC_INPUT02_CLK	Primary Oscillator Clock	SYS_CLK	0x2
DCC_CLKSRC0_CLK	Counter 1 Clock Source	PLL_DSS_HSDIV0_CLKOUT2	0x0
DCC_CLKSRC3_CLK	Counter 1 Clock Source	GPIO[0]	0x3
DCC_CLKSRC4_CLK	Counter 1 Clock Source	GPIO[1]	0x4
DCC_CLKSRC5_CLK	Counter 1 Clock Source	GPIO[2]	0x5
DCC_CLKSRC6_CLK	Counter 1 Clock Source	MSS_WDT_CLK	0x6
DCC_CLKSRC7_CLK	Counter 1 Clock Source	XTALCLK	0x7

**Table 20-5. MSS\_DCCD Clocking**

Clock Signal	Description	Source	REG Value for Selection
DCC_INPUT00_CLK	Primary Oscillator Clock	XTALCLK	0x0
DCC_INPUT01_CLK	Primary Oscillator Clock	RCCLK10	0x1
DCC_INPUT02_CLK	Primary Oscillator Clock	SYS_CLK	0x2
DCC_CLKSRC0_CLK	Counter 1 Clock Source	PLL_PER_HSDIV0_CLKOUT1	0x0
DCC_CLKSRC1_CLK	Counter 1 Clock Source	MSS_WDT_CLK	0x1
DCC_CLKSRC2_CLK	Counter 1 Clock Source	MSS_MCANA_CLK	0x2
DCC_CLKSRC3_CLK	Counter 1 Clock Source	GPIO[8]	0x3
DCC_CLKSRC4_CLK	Counter 1 Clock Source	GPIO[9]	0x4
DCC_CLKSRC5_CLK	Counter 1 Clock Source	GPIO[10]	0x5
DCC_CLKSRC6_CLK	Counter 1 Clock Source	GPIO[0]	0x6
DCC_CLKSRC7_CLK	Counter 1 Clock Source	GPIO[1]	0x7

## 20.2.4 Clock Source Selection for Counter0 and Counter1

Refer the device datasheet to identify the available options for selecting the clock sources for both counters of the DCC module. Some microcontrollers may include multiple instances of the DCC module. This will also be identified in the device datasheet.

The selection of the clock sources for counter0 and coutner1 is done by a combination of the KEY, CNT0 CLKSRC and CNT1 CLKSRC control fields of the CNT0CLKSRC and CNT1CLKSRC registers.

## 20.2.5 DCC Registers

Table 20-6 lists the DCC memory-mapped registers. All register offset addresses not listed in Table 20-6 should be considered as reserved locations and the register contents should not be modified.

**Table 20-6. DCC Registers**

Offset	Acronym	Register Name	Section
0h	DCCGCTRL	DCCGCTRL	<a href="#">Section 20.2.5.1</a>
4h	DCCREV	DCCREV	<a href="#">Section 20.2.5.2</a>
8h	DCCCNTSEED0	DCCCNTSEED0	<a href="#">Section 20.2.5.3</a>
Ch	DCCVALIDSEED0	DCCVALIDSEED0	<a href="#">Section 20.2.5.4</a>
10h	DCCCNTSEED1	DCCCNTSEED1	<a href="#">Section 20.2.5.5</a>
14h	DCCSTAT	DCCSTAT	<a href="#">Section 20.2.5.6</a>
18h	DCCCNT0	DCCCNT0	<a href="#">Section 20.2.5.7</a>
1Ch	DCCVALID0	DCCVALID0	<a href="#">Section 20.2.5.8</a>
20h	DCCCNT1	DCCCNT1	<a href="#">Section 20.2.5.9</a>
24h	DCCCLKSSRC1	DCCCLKSSRC1	<a href="#">Section 20.2.5.10</a>
28h	DCCCLKSSRC0	DCCCLKSSRC0	<a href="#">Section 20.2.5.11</a>

Complex bit access types are encoded to fit into small table cells. Table 20-7 shows the codes that are used for access types in this section.

**Table 20-7. DCC Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value
Register Array Variables		
i,j,k,l,m,n		When these variables are used in a register name, an offset, or an address, they refer to the value of a register array where the register is part of a group of repeating registers. The register groups form a hierarchical structure and the array is represented with a formula.
y		When this variable is used in a register name, an offset, or an address it refers to the value of a register array.

### 20.2.5.1 DCCGCTRL Register (Offset = 0h) [reset = 5555h]

DCCGCTRL is shown in [Figure 20-8](#) and described in [Table 20-8](#).

Return to [Summary Table](#).

Starts / stops the counters clears the error signal

**Figure 20-8. DCCGCTRL Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DONENA				SINGLESOT				ERRENA				DCCENA			
R/W-5h				R/W-5h				R/W-5h				R/W-5h			

**Table 20-8. DCCGCTRL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	Reserved
15-12	DONENA	R/W	5h	The DONEENA bit enables/disables the done signal. 0101 = disabled & 1010 = enabled
11-8	SINGLESOT	R/W	5h	Single/Continuous checking mode. 0101 = Continuous & 1010 = Single
7-4	ERRENA	R/W	5h	The ERRENA bit enables/disables the error signal. 0101 = disabled & 1010 = enabled
3-0	DCCENA	R/W	5h	The DCCENA bit starts and stops the operation of the dcc 0101 = disabled & 1010 = enabled



### 20.2.5.2 DCCREV Register (Offset = 4h) [reset = 4000204h]

DCCREV is shown in [Figure 20-9](#) and described in [Table 20-9](#).

Return to [Summary Table](#).

Module version

**Figure 20-9. DCCREV Register**

31	30	29	28	27	26	25	24
NU2	SCHEME			NU1		FUNC	
R-0h	R-4h			R-0h		R-0h	
23	22	21	20	19	18	17	16
FUNC							
R-0h							
15	14	13	12	11	10	9	8
FUNC		RTL					MAJOR
R-0h		R-1h					R-0h
7	6	5	4	3	2	1	0
MAJOR		CUSTOM	MINOR				
R-0h		R-0h	R-4h				

**Table 20-9. DCCREV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NU2	R	0h	Reserved
30-28	SCHEME	R	4h	SCHEME. - (RO )
27-26	NU1	R	0h	Reserved
25-14	FUNC	R	0h	Functional release number - (RO )
13-9	RTL	R	1h	Design Release Number - (RO )
8-6	MAJOR	R	0h	Major Revision Number - (RO )
5	CUSTOM	R	0h	Indicates a special version of the module. May not be supported by standard software - (RO )
4-0	MINOR	R	4h	Minor revision number. - (RO )

### 20.2.5.3 DCCCNTSEED0 Register (Offset = 8h) [reset = 0h]

DCCCNTSEED0 is shown in [Figure 20-10](#) and described in [Table 20-10](#).

Return to [Summary Table](#).

Seed value for the counter attached to clock source 0

**Figure 20-10. DCCCNTSEED0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU3												COUNTSEED0																			
R-0h												R/W-0h																			

**Table 20-10. DCCCNTSEED0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU3	R	0h	Reserved
19-0	COUNTSEED0	R/W	0h	The seed value for Counter 0. The seed value that gets loaded into counter 0 (clock source 0)

#### 20.2.5.4 DCCVALIDSEED0 Register (Offset = Ch) [reset = 0h]

DCCVALIDSEED0 is shown in [Figure 20-11](#) and described in [Table 20-11](#).

Return to [Summary Table](#).

Seed value for the timeout counter attached to clock source 0

**Figure 20-11. DCCVALIDSEED0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU4																VALIDSEED0															
R-0h																R/W-0h															

**Table 20-11. DCCVALIDSEED0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU4	R	0h	Reserved
15-0	VALIDSEED0	R/W	0h	The seed value for Valid Duration Counter 0. The seed value that gets loaded into the valid duration counter for clock source 0

### 20.2.5.5 DCCCNTSEED1 Register (Offset = 10h) [reset = 0h]

DCCCNTSEED1 is shown in [Figure 20-12](#) and described in [Table 20-12](#).

Return to [Summary Table](#).

Seed value for the counter attached to clock source 1

**Figure 20-12. DCCCNTSEED1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU5												COUNTSEED1																			
R-0h												R/W-0h																			

**Table 20-12. DCCCNTSEED1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU5	R	0h	Reserved
19-0	COUNTSEED1	R/W	0h	The seed value for Counter 1. The seed value that gets loaded into counter 1 (clock source 1

### 20.2.5.6 DCCSTAT Register (Offset = 14h) [reset = 0h]

DCCSTAT is shown in [Figure 20-13](#) and described in [Table 20-13](#).

Return to [Summary Table](#).

Contains the error & done flag bit

**Figure 20-13. DCCSTAT Register**

31	30	29	28	27	26	25	24
NU6							
R-0h							
23	22	21	20	19	18	17	16
NU6							
R-0h							
15	14	13	12	11	10	9	8
NU6							
R-0h							
7	6	5	4	3	2	1	0
NU6						DONE	ERR
R-0h						R/W-0h	R/W-0h

**Table 20-13. DCCSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU6	R	0h	Reserved
1	DONE	R/W	0h	Indicates whether or not an done has occurred. Writing a 1 to this bit clears the flag.
0	ERR	R/W	0h	Indicates whether or not an error has occurred. Writing a 1 to this bit clears the flag.

### 20.2.5.7 DCCCNT0 Register (Offset = 18h) [reset = 0h]

DCCCNT0 is shown in [Figure 20-14](#) and described in [Table 20-14](#).

Return to [Summary Table](#).

Value of the counter attached to clock source 0

**Figure 20-14. DCCCNT0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU7												COUNT0																			
R-0h												R-0h																			

**Table 20-14. DCCCNT0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU7	R	0h	Reserved
19-0	COUNT0	R	0h	This field contains the current value of counter 0. - (RO )

### 20.2.5.8 DCCVALID0 Register (Offset = 1Ch) [reset = 0h]

DCCVALID0 is shown in [Figure 20-15](#) and described in [Table 20-15](#).

Return to [Summary Table](#).

Value of the valid counter attached to clock source 0

**Figure 20-15. DCCVALID0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU8																VALID0															
R-0h																R-0h															

**Table 20-15. DCCVALID0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU8	R	0h	Reserved
15-0	VALID0	R	0h	This field contains the current value of valid counter 0. - (RO )

### 20.2.5.9 DCCCNT1 Register (Offset = 20h) [reset = 0h]

DCCCNT1 is shown in [Figure 20-16](#) and described in [Table 20-16](#).

Return to [Summary Table](#).

Value of the counter attached to clock source 1

**Figure 20-16. DCCCNT1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU9												COUNT1																			
R-0h												R-0h																			

**Table 20-16. DCCCNT1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU9	R	0h	Reserved
19-0	COUNT1	R	0h	This field contains the current value of counter 1. - (RO )



### 20.2.5.10 DCCCLKSSRC1 Register (Offset = 24h) [reset = 5000h]

DCCCLKSSRC1 is shown in [Figure 20-17](#) and described in [Table 20-17](#).

Return to [Summary Table](#).

Clock source1 selection control

**Figure 20-17. DCCCLKSSRC1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU11															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
KEY_B4				NU10								CLK_SRC1			
R/W-5h								R-0h				R/W-0h			

**Table 20-17. DCCCLKSSRC1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU11	R	0h	Reserved
15-12	KEY_B4	R/W	5h	Key Programming (1010 is the KEY Value)
11-4	NU10	R	0h	Reserved
3-0	CLK_SRC1	R/W	0h	Clock source selection for Source 0 DCC-A Clock source-0 selection Program value and its respective clock selected 0x0 - REF_CLK 0x1 - CPU_CLK 0x2 - RC_CLK 0x3 - RC_CLK 0x4 - RC_CLK 0x5 - RC_CLK 0x6 - RC_CLK 0x7 - RC_CLK DCC-B Clock source-0 selection Program value and its respective clock selected 0x0 - VCLK 0x1 - DSS_CLK 0x2 - BSS_CLK 0x3 - QSPI_CLK 0x4 - FDCAN_CLK 0x5 - RED_CLK 0x6 - CPU_CLK 0x7 - RC_CLK

### 20.2.5.11 DCCCLKSSRC0 Register (Offset = 28h) [reset = 5h]

DCCCLKSSRC0 is shown in [Figure 20-18](#) and described in [Table 20-18](#).

Return to [Summary Table](#).

Clock source0 selection control

**Figure 20-18. DCCCLKSSRC0 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU12															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU12												CLK_SRC0			
R-0h												R/W-5h			

**Table 20-18. DCCCLKSSRC0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	NU12	R	0h	Reserved
3-0	CLK_SRC0	R/W	5h	Clock source selection for Source 0 DCC-A Clock source-0 selection Program value and its respective clock selected 0 - REF_CLK A - PLL_600 5 - PLL_240 DCC-B Clock source-0 selection Program value and its respective clock selected 0 - PLL_600 A - VCLK 5 - CPU_CLK

## 20.3 ECC\_AGGREGATOR

This section describes the common ECC aggregator functionality.

### 20.3.1 ECC Aggregator Overview

To increase functional safety and system reliability the memories (for example, FIFOs, queues, SRAMs and others) in many device modules and subsystems are protected by error correcting code (ECC). This is accomplished through an ECC aggregator and ECC wrapper. The ECC aggregator is connected to these memories (hereinafter ECC RAMs) and involved in the ECC process. Each memory is surrounded by an ECC wrapper which performs the ECC detection and correction. The wrapper communicates via serial interface with the aggregator which has memory mapped configuration interface

### 20.3.2 Integration Details

In AWR2544 design, there are four ECC aggregators.

- MSS\_ECC\_AGG\_R5A
- MSS\_ECC\_AGG\_MSS
- DSS\_ECC\_AGG
- HSM\_ECC\_AGG (details will be available in HSM document)

This aggregator is used to fault inject all memory ecc\_controllers and aggregate the errors to generate a single error to ESM.

### 20.3.3 ECC Aggregator Features

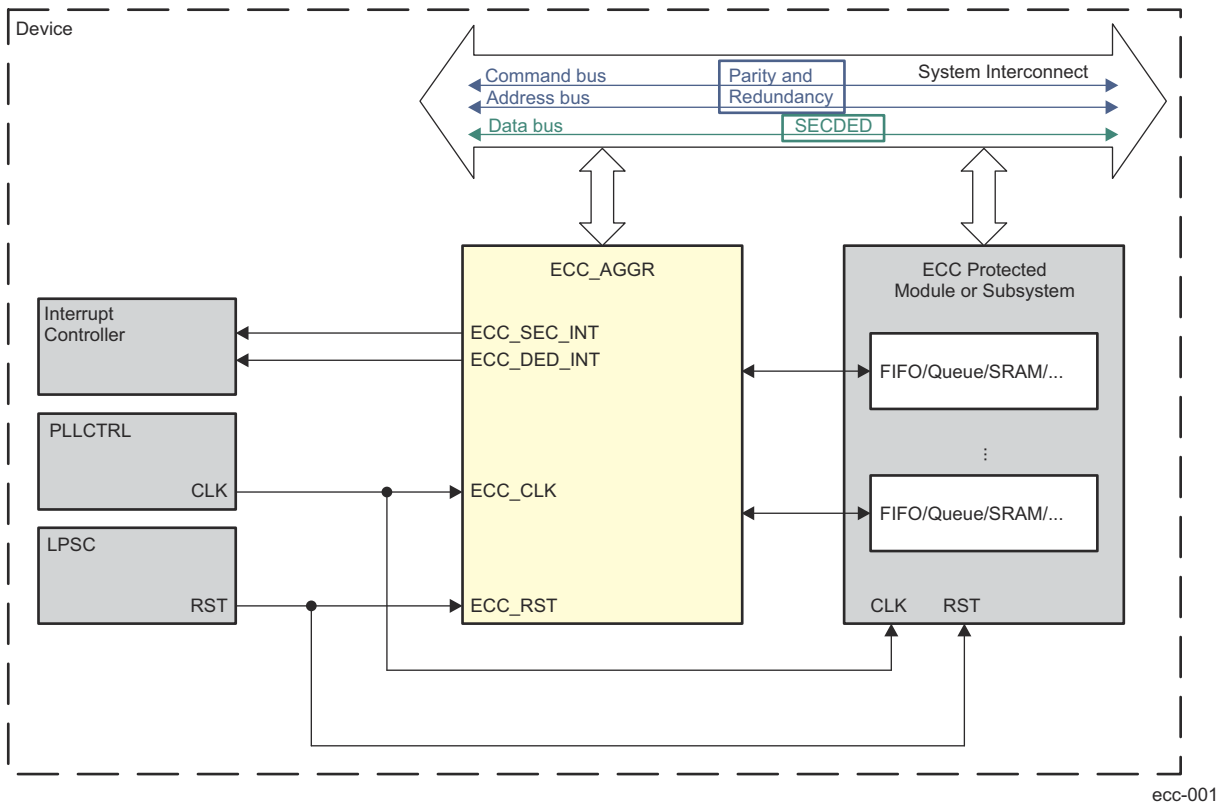
The ECC aggregator has the following features:

- Reduces memory software errors via single error correction (SEC) and double error detection (DED)
- Provides a mechanism to control and monitor the ECC protected memories in a module or subsystem
- SEC and DED over the system interconnect data bus and parity and redundancy for the system interconnect command and address buses
- Generates an interrupt for correctable error

- Generates an interrupt for non-correctable error
- Supports inject only mode for diagnostic purposes
- Supports software readable status for single and double-bit ECC errors and associated information such as row address where error has occurred and data bits that have been flipped
- An ECC endpoint can be ECC RAM component.
- Detects single bit error via parity checking on:
  1. Memory mapped configuration interface FIFO
  2. Serial interface FIFO
- Single bit error detection via parity checking results in a non-correctable error interrupt
- Supports timeout mechanism on transactions over the ECC serial interface. Timeout occurrence results in a non-correctable error interrupt.
- Certain control bits have redundancy and if a bit flips an interrupt is generated

### 20.3.4 ECC Aggregator Integration

This section describes ECC aggregator integration in the device, including information about clocks, resets, and hardware requests.



**Figure 20-19. ECC Aggregator Integration**

**Table 20-19. ECC Aggregator Clocks and Resets**

Clock				
Module Instance	Module Clock Input	Source Clock Signal	Source	Description
ECC_AGGR	ECC_CLK	Same as corresponding module or subsystem	Same as corresponding module or subsystem	ECC aggregator clock
Resets				
Module Instance	Module Reset Input	Source Reset Signal	Source	Description
ECC_AGGR	ECC_RST	Same as corresponding module or subsystem	Same as corresponding module or subsystem	ECC aggregator reset

**Table 20-20. ECC Aggregator Hardware Requests**

Interrupt Requests					
Module Instance	Module Interrupt Signal	Destination Interrupt Input	Description	Description	Type
ECC_AGGR	ECC_SEC_INT	See	See	Interrupt for correctable error (SEC)	Level
	ECC_DED_INT	See	See	Interrupt for non-correctable error (DED, parity, redundancy, timeout)	LEVEL
DMA Events					
Module Instance	Module DMA Input	Destination DMA Event Input	Destination	Description	Type
ECC_AGGR	-	-	-	-	-

### Note

For more information on the interrupts, see *Interrupts*.

For more information on the interconnects, see *System Interconnect*.

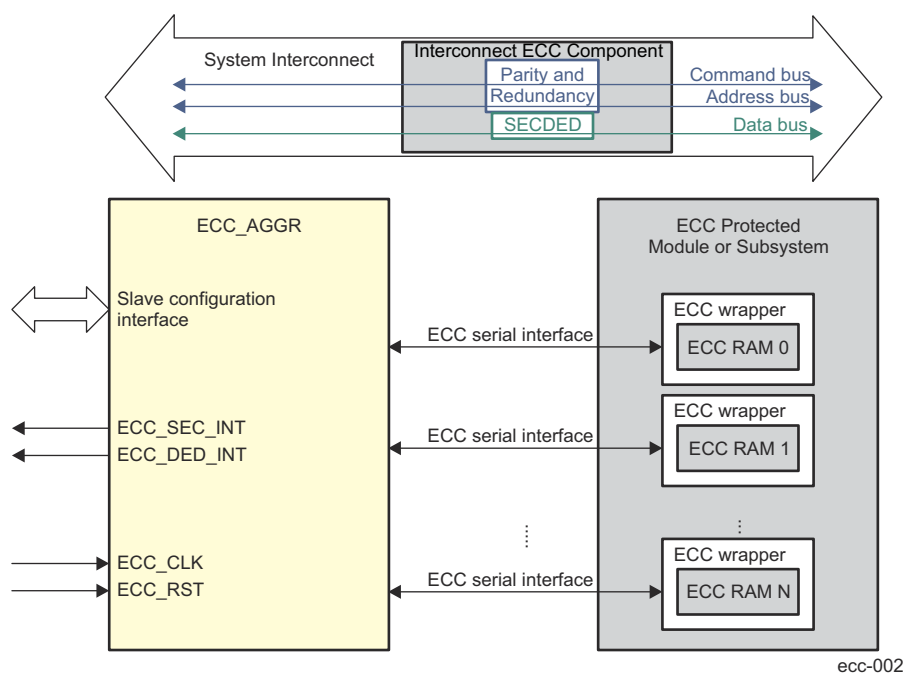
For more information on the power, reset and clock management, see the corresponding sections in *Device Configuration*.

For more information on the device interrupt controllers, see *Interrupt Controllers*.

## 20.3.5 ECC Aggregator Function Description

This section describes the architecture and functional details of the ECC aggregator.

### 20.3.5.1 ECC Aggregator Block Diagram


**Figure 20-20. ECC Aggregator Block Diagram**

The ECC aggregator is connected to one or more ECC endpoints each of which has assigned a unique ID used when the endpoint is accessed for status information or configuration. The ECC aggregator provides software access to all ECC related registers through its memory mapped slave configuration interface while the serial interface is used to communicate with the ECC endpoints. Upon detection of single or double-bit error the corresponding interrupt line is asserted.

### 20.3.6 ECC Aggregator Register Groups

The ECC aggregator has ECC control, status and interrupt registers for each ECC endpoint in a module or subsystem. These registers are memory mapped and occupy 1 KB address space although part of it may contain reserved locations. The registers are split in the following types:

- **Global registers.** They are common to all ECC endpoints associated with the ECC aggregator and include the ECC\_VECTOR and ECC\_REV registers. Each ECC endpoint has assigned a unique ID.

When this ID is written to the ECC\_VECTOR[10-0] ECC\_VECTOR field the corresponding endpoint is selected either for control or for status reading.

- **ECC control and status registers.** These registers are specific to each ECC endpoint and reside in the range from address offset 0x10 to 0x28, if the endpoint is ECC RAM or from 0x10 to 0x24, if the endpoint is interconnect ECC component. They are memory mapped but are accessed through the ECC serial interface. They are also selected by the ECC endpoint ID written to the ECC\_VECTOR[10-0] ECC\_VECTOR field. Because of latency on the serial interface the ECC control and status registers are read by performing special sequence as described in Section 12.9.4.3.3. These registers have also different functionality for both types of endpoints - ECC RAM and interconnect ECC component.
- **Interrupt registers.** They include interrupt status, interrupt enable, interrupt disable, and EOI registers.

### 20.3.7 Read Access to the ECC Control and Status Registers

Read accesses to the ECC control and status registers for each ECC endpoint represent read operations over the ECC serial interface and are triggered by performing the following sequence:

1. Software writes the following in the ECC\_VECTOR register:
  - The ECC endpoint ID in the ECC\_VECTOR[10-0] ECC\_VECTOR field to select particular ECC endpoint.
  - The register read address in the ECC\_VECTOR[23-16] RD\_SVBUS\_ADDRESS field to select which register has to be read through the ECC serial interface.
  - A value of 0x1 in the ECC\_VECTOR[15] RD\_SVBUS bit to trigger read operation through the ECC serial interface.
2. Software polls the ECC\_VECTOR[24] RD\_SVBUS\_DONE bit to check if it is 0x1. This indicates that the read operation on the ECC serial interface has completed.
3. Software reads the data from the register previously selected by the ECC\_VECTOR[23-16]RD\_SVBUS\_ADDRESS field.

### 20.3.8 Serial Write Operation

Write operations over the ECC serial interface are performed as follows:

1. Software specifies the ECC endpoint ID in the ECC\_VECTOR[10-0] ECC\_VECTOR field. The ECC\_VECTOR[23-16] RD\_SVBUS\_ADDRESS field is a don't care but the ECC\_VECTOR[15] RD\_SVBUS bit must be set to 0x0.
2. Software performs regular write operation to the desired address. If the ECC endpoint ID has already been specified, step 1 can be skipped. Unlike serial read operations it is not necessary to always specify the endpoint ID before performing serial write operation.

The following is an example for serial write operation:

1. Write 0x0000 0008 to the ECC\_VECTOR register.
2. Write 0x0000 000F to the ECC\_CTRL register. This sends write request with data 0x0000 000F to the ECC\_CTRL register associated with ECC RAM with ID = 8.

### 20.3.9 Interrupts

The ECC aggregator generates the following interrupts:

- Correctable interrupt (ECC\_SEC\_INT) where hardware can correct the error but notifies the system in case of SEC.
- Non-correctable interrupt (ECC\_DED\_INT) where hardware cannot correct the error in cases of DED, parity check, redundancy check or timeout occurrence.

The following is the sequence for servicing interrupts:

- Software enables the interrupts for an ECC endpoint by writing 0x1 to the corresponding bit of the following interrupt enable registers:
  - ECC\_SEC\_ENABLE\_SET\_REG0 for the correctable interrupt
  - ECC\_DED\_ENABLE\_SET\_REG0 for the noncorrectable interrupt
- On receiving an interrupt, software checks which ECC endpoint has caused the error by reading the following interrupt status registers:
  - ECC\_SEC\_STATUS\_REG0 for the correctable interrupt ECC\_DED\_STATUS\_REG0 for the non-correctable interrupt
- Software performs serial read operations as described in Section 12.9.4.3.3 to read the following status registers that contain details about the error:
  - If the endpoint is ECC RAM:
    - ECC\_ERR\_STAT1
    - ECC\_ERR\_STAT2
    - ECC\_ERR\_STAT3
  - If the endpoint is interconnect ECC component:
    - ECC\_CBASS\_ERR\_STAT1
    - ECC\_CBASS\_ERR\_STAT2
- After the interrupt has been serviced, depending on the error type, software should clear the corresponding status bits in the ECC\_ERR\_STAT1 and ECC\_ERR\_STAT3 registers or in the ECC\_CBASS\_ERR\_STAT1 register. Software has to poll these registers to guarantee that status bits are cleared as there is no other indication for write completion over the ECC serial interface.

The value of the \*\_PEND\_CLR fields in the ECC\_CBASS\_ERR\_STAT1 register must be read and then written back to decrement the count of each field back to 0x0. A further error capture into the ECC\_CBASS\_ERR\_STAT1 register does not occur unless all its fields are 0x0. The decrement value should not be larger than the read value. If a field in the ECC\_CBASS\_ERR\_STAT1 register should not be modified, write a value of 0x0 to that field.

- Software writes 0x1 to the corresponding end of interrupt register to clear the interrupt:
  - ECC\_SEC\_EOI\_REG for the correctable interrupt
  - ECC\_DED\_EOI\_REG for the non-correctable interrupt

#### Note

For AWR2544 device the ECC aggregator is used only for the memories and not on interconnects. Hence any register set and text for the interconnect ECC will not be applicable to the device and can be ignored.

### 20.3.10 Inject Only Mode

There are modules that already perform the ECC generation and checking as part of their data path. In this case, the ECC wrapper may be configured in inject only mode, if needed. In this mode the ECC wrapper does not perform ECC detection and correction. The inject only mode allows users to inject single or double-bit errors so that the module logic can be tested for diagnostic purposes.

### Note

There is no software control to enable inject only mode. It is configured via tie-off value. Inject only and ECC modes are mutually exclusive.

The interconnect ECC component also supports error injection mode. There is error injection logic for testing of the error checking logic (checkers). The injection logic can be configured to inject either single or double bit error and what data pattern to be used for injection (ECC\_CBASS\_CTRL[11-8] ECC\_PATTERN). The ECC\_CBASS\_ERR\_CTRL1 and ECC\_CBASS\_ERR\_CTRL2 registers should be written first to setup the injection. Then, either the ECC\_CBASS\_CTRL[3] FORCE\_SE or the ECC\_CBASS\_CTRL[4] FORCE\_DE bit must be set to 0x1 to start the injection. Both bits must not be set at the same time. If the injection should continue in incrementing mode, then the ECC\_CBASS\_CTRL[5] FORCE\_N\_BIT bit should be set to 0x1. Once the FORCE\_N\_BIT is set, then each successive injection can simply write the ECC\_CBASS\_CTRL register to set the FORCE\_SE or FORCE\_DE again. Reading 0x0 from either the FORCE\_SE or the FORCE\_DE bit indicates that the injection has completed, as these bits automatically clear when the checker indicates that it has performed the injection. The time for an injection to complete is not guaranteed, so some delay is needed between successive injections.

### Note

For AWR2544 device the ECC aggregator is used only for the memories and not on interconnects. Hence any register set and text for the interconnect ECC will not be applicable to the device and can be ignored.

## 20.3.11 Errors

Each aggregator generates two errors which drive the ESM.

- <modulename>\_SERR module names are mentioned in the below section
- <modulename>\_SERR module names are mentioned in the below section

Group1 and Group2 mappings are found the ESM interrupt sections.

## 20.3.12 Aggregator Mapping to Memory Instances

**Table 20-21. MSS\_ECC\_AGG\_R5A Instance**

RAM ID	Module Name	Protected RAM
--------	-------------	---------------

**Table 20-21. MSS\_ECC\_AGG\_R5A Instance (continued)**

RAM_0	MSS_CR5A_CACHE	MSS_CR5A_ITAG_BANK0
RAM_1		MSS_CR5A_ITAG_BANK1
RAM_2		MSS_CR5A_ITAG_BANK2
RAM_3		MSS_CR5A_ITAG_BANK3
RAM_4		MSS_CR5A_IDATA_BANK0
RAM_5		MSS_CR5A_IDATA_BANK1
RAM_6		MSS_CR5A_IDATA_BANK2
RAM_7		MSS_CR5A_IDATA_BANK3
RAM_8		MSS_CR5A_DTAG_BANK0
RAM_9		MSS_CR5A_DTAG_BANK1
RAM_10		MSS_CR5A_DTAG_BANK2
RAM_11		MSS_CR5A_DTAG_BANK3
RAM_12		MSS_CR5A_DDIRTY
RAM_13		MSS_CR5A_DDATA_BANK0
RAM_14		MSS_CR5A_DDATA_BANK1
RAM_15		MSS_CR5A_DDATA_BANK2
RAM_16		MSS_CR5A_DDATA_BANK3
RAM_17		MSS_CR5A_DDATA_BANK4
RAM_18		MSS_CR5A_DDATA_BANK5
RAM_19		MSS_CR5A_DDATA_BANK6
RAM_20		MSS_CR5A_DDATA_BANK7
RAM_21	MSS_CR5A_TCM	MSS_CR5A_ATCM_BANK0
RAM_22		MSS_CR5A_ATCM_BANK1
RAM_23		MSS_CR5A_B0TCM_BANK0
RAM_24		MSS_CR5A_B0TCM_BANK1
RAM_25		MSS_CR5A_B1TCM_BANK0
RAM_26		MSS_CR5A_B1TCM_BANK1
RAM_27	MSS_CR5A_VIM	MSS_CR5A_VIM

**Table 20-22. MSS\_ECC\_AGG\_MSS Instance**

RAM ID	Module Name	Protected RAM
RAM_0	MSS_L2	MSS_L2RAM A
RAM_1		MSS_L2RAM B
RAM_2	MSS_MBOX	MSS_MBOX
RAM_3	MSS_RETRAM	MSS_RETRAM
RAM_4	MSS_GPADC	MSS_GPADC_DATA_RAM
RAM_5	MSS_TPTC_A0	MSS_TPTC_A0
RAM_6	MSS_TPTC_A1	MSS_TPTC_A1
RAM_7	MSS_L2	MSS_L2RAM C

**Table 20-23. DSS\_ECC\_AGG Instance**

RAM ID	Module Name	Protected RAM
RAM_0	DSS_L3	DSS_L3RAM A
RAM_1		DSS_L3RAM B
RAM_9	DSS_TPTC_A0	DSS TPTC A0 FIFO
RAM_10	DSS_TPTC_A1	DSS TPTC A1 FIFO



**Table 20-23. DSS\_ECC\_AGG Instance (continued)**

RAM_21	DSS_HWA	DSS HWA PARAM RAM
--------	---------	-------------------

## 20.4 ECC Registers

### 20.4.1 DSS\_ECC\_AGG Registers

Table 20-24 lists the memory-mapped registers for the DSS\_ECC\_AGG registers. All register offset addresses not listed in Table 20-24 should be considered as reserved locations and the register contents should not be modified.

**Table 20-24. DSS\_ECC\_AGG Registers**

Offset	Acronym	Register Name	Section
0h	rev	Aggregator Revision Register	<a href="#">Go</a>
8h	vector	ECC Vector Register	<a href="#">Go</a>
Ch	stat	Misc Status	<a href="#">Go</a>
10h	wrap_rev	ECC Wrapper Revision Register	<a href="#">Go</a>
14h	ctrl	ECC Control	<a href="#">Go</a>
18h	err_ctrl1	ECC Error Control1 Register	<a href="#">Go</a>
1Ch	err_ctrl2	ECC Error Control2 Register	<a href="#">Go</a>
20h	err_stat1	ECC Error Status1 Register	<a href="#">Go</a>
24h	err_stat2	ECC Error Status2 Register	<a href="#">Go</a>
28h	err_stat3	ECC Error Status3 Register	<a href="#">Go</a>
3Ch	sec_eoi_reg	EOI Register	<a href="#">Go</a>
40h	sec_status_reg0	Interrupt Status Register 0	<a href="#">Go</a>
80h	sec_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Go</a>
C0h	sec_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Go</a>
13Ch	ded_eoi_reg	EOI Register	<a href="#">Go</a>
140h	ded_status_reg0	Interrupt Status Register 0	<a href="#">Go</a>
180h	ded_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Go</a>
1C0h	ded_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Go</a>
200h	aggr_enable_set	AGGR interrupt enable set Register	<a href="#">Go</a>
204h	aggr_enable_clr	AGGR interrupt enable clear Register	<a href="#">Go</a>
208h	aggr_status_set	AGGR interrupt status set Register	<a href="#">Go</a>
20Ch	aggr_status_clr	AGGR interrupt status clear Register	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. Table 20-25 shows the codes that are used for access types in this section.

**Table 20-25. DSS\_ECC\_AGG Access Type Codes**

Access Type	Code	Description
<b>Read Type</b>		
R	R	Read
<b>Write Type</b>		
W	W	Write
W1C	W 1C	Write 1 to clear
W1S	W 1S	Write 1 to set
Wdecr	W decr	Write
Wincr	W incr	Write
<b>Reset or Default Value</b>		

**Table 20-25. DSS\_ECC\_AGG Access Type Codes  
(continued)**

Access Type	Code	Description
$-n$		Value after reset or the default value

### 20.4.1.1 rev Register (Offset = 0h) [Reset = 66A0C200h]

rev is shown in [Table 20-26](#).

Return to the [Summary Table](#).

Revision parameters

**Table 20-26. rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A0h	Module ID
15-11	revrtl	R	18h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	0h	Minor version

### 20.4.1.2 vector Register (Offset = 8h) [Reset = X]

vector is shown in [Table 20-27](#).

Return to the [Summary Table](#).

ECC Vector Register

**Table 20-27. vector Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	X	
24	rd_svbus_done	R	0h	Status to indicate if read on serial VBUS is complete
23-16	rd_svbus_address	R/W	0h	Read address
15	rd_svbus	R/W1S	0h	Write 1 to trigger a read on the serial VBUS
14-11	RESERVED	R/W	X	
10-0	ecc_vector	R/W	0h	Value written to select the corresponding ECC RAM for control or status

### 20.4.1.3 stat Register (Offset = Ch) [Reset = X]

stat is shown in [Table 20-28](#).

Return to the [Summary Table](#).

Misc Status

**Table 20-28. stat Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	X	
10-0	num_rams	R	17h	Indicates the number of RAMS serviced by the ECC aggregator

#### 20.4.1.4 wrap\_rev Register (Offset = 10h) [Reset = 66A40202h]

wrap\_rev is shown in [Table 20-29](#).

Return to the [Summary Table](#).

Revision parameters

**Table 20-29. wrap\_rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A4h	Module ID
15-11	revrtl	R	0h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	2h	Minor version

### 20.4.1.5 ctrl Register (Offset = 14h) [Reset = X]

ctrl is shown in [Table 20-30](#).

Return to the [Summary Table](#).

ECC Control Register

**Table 20-30. ctrl Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	check_svbus_timeout	R/W	1h	check for svbus timeout errors
7	check_parity	R/W	1h	check for parity errors
6	error_once	R/W	0h	Force Error only once
5	force_n_row	R/W	0h	Force Error on any RAM read
4	force_ded	R/W	0h	Force Double Bit Error
3	force_sec	R/W	0h	Force Single Bit Error
2	enable_rmw	R/W	1h	Enable rmw
1	ecc_check	R/W	1h	Enable ECC check
0	ecc_enable	R/W	1h	Enable ECC



### 20.4.1.6 err\_ctrl1 Register (Offset = 18h) [Reset = 00000000h]

err\_ctrl1 is shown in [Table 20-31](#).

Return to the [Summary Table](#).

ECC Error Control1 Register

**Table 20-31. err\_ctrl1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R/W	0h	Row address where single or double-bit error needs to be applied. This is ignored if force_n_row is set

### 20.4.1.7 err\_ctrl2 Register (Offset = 1Ch) [Reset = 00000000h]

err\_ctrl2 is shown in [Table 20-32](#).

Return to the [Summary Table](#).

ECC Error Control2 Register

**Table 20-32. err\_ctrl2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit2	R/W	0h	Data bit that needs to be flipped if double bit error needs to be forced
15-0	ecc_bit1	R/W	0h	Data bit that needs to be flipped when force_sec is set

### 20.4.1.8 err\_stat1 Register (Offset = 20h) [Reset = 00000000h]

err\_stat1 is shown in [Table 20-33](#).

Return to the [Summary Table](#).

ECC Error Status1 Register

**Table 20-33. err\_stat1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit1	R	0h	Data bit that corresponds to the single-bit error
15	clr_ctrl_reg_err	R/W1C	0h	Clear control reg error Error Status, you must also re write the control register itself to clear this
14-13	clr_parity_err	R/Wdecr	0h	Clear parity Error Status
12	clr_ecc_other	R/W1C	0h	Clear other Error Status
11-10	clr_ecc_ded	R/Wdecr	0h	Clear Double Bit Error Status
9-8	clr_ecc_sec	R/Wdecr	0h	Clear Single Bit Error Status
7	ctr_reg_err	R/W1S	0h	control register error pending, Level interrupt
6-5	parity_err	R/W1S	0h	Level parity error Error Status
4	ecc_other	R/W1S	0h	successive single-bit errors have occurred while a writeback is still pending, Level interrupt
3-2	ecc_ded	R/Wincr	0h	Level Double Bit Error Status
1-0	ecc_sec	R/Wincr	0h	Level Single Bit Error Status

### 20.4.1.9 err\_stat2 Register (Offset = 24h) [Reset = 00000000h]

err\_stat2 is shown in [Table 20-34](#).

Return to the [Summary Table](#).

ECC Error Status2 Register

**Table 20-34. err\_stat2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R	0h	Row address where the single or double-bit error has occurred

#### 20.4.1.10 err\_stat3 Register (Offset = 28h) [Reset = X]

err\_stat3 is shown in [Table 20-35](#).

Return to the [Summary Table](#).

ECC Error Status3 Register

**Table 20-35. err\_stat3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	
9	clr_svbus_timeout_err	R/W1C	0h	Clear svbus timeout Error Status
8-2	RESERVED	R/W	X	
1	svbus_timeout_err	R/W1S	0h	Level svbus timeout error Error Status
0	wb_pend	R	0h	delayed write back pending Status

### 20.4.1.11 sec\_eoi\_reg Register (Offset = 3Ch) [Reset = X]

sec\_eoi\_reg is shown in [Table 20-36](#).

Return to the [Summary Table](#).

EOI Register

**Table 20-36. sec\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

### 20.4.1.12 sec\_status\_reg0 Register (Offset = 40h) [Reset = X]

sec\_status\_reg0 is shown in [Table 20-37](#).

Return to the [Summary Table](#).

Interrupt Status Register 0

**Table 20-37. sec\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22	rcss_tptc_b1_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_b1_pend
21	rcss_tptc_b0_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_b0_pend
20	rcss_tptc_a1_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_a1_pend
19	rcss_tptc_a0_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_a0_pend
18	dss_tptc_c5_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c5_pend
17	dss_tptc_c4_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c4_pend
16	dss_tptc_c3_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c3_pend
15	dss_tptc_c2_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c2_pend
14	dss_tptc_c1_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c1_pend
13	dss_tptc_c0_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c0_pend
12	dss_tptc_b1_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_b1_pend
11	dss_tptc_b0_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_b0_pend
10	dss_tptc_a1_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_a1_pend
9	dss_tptc_a0_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_a0_pend
8-5	RESERVED	R/W	0h	
4	dss_mailbox_pend	R/W1S	0h	Interrupt Pending Status for dss_mailbox_pend
3	dss_l3ram3_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram3_pend
2	dss_l3ram2_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram2_pend
1	dss_l3ram1_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram1_pend
0	dss_l3ram0_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram0_pend

### 20.4.1.13 sec\_enable\_set\_reg0 Register (Offset = 80h) [Reset = X]

sec\_enable\_set\_reg0 is shown in [Table 20-38](#).

Return to the [Summary Table](#).

Interrupt Enable Set Register 0

**Table 20-38. sec\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22	rcss_tptc_b1_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_b1_pend
21	rcss_tptc_b0_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_b0_pend
20	rcss_tptc_a1_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_a1_pend
19	rcss_tptc_a0_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_a0_pend
18	dss_tptc_c5_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c5_pend
17	dss_tptc_c4_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c4_pend
16	dss_tptc_c3_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c3_pend
15	dss_tptc_c2_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c2_pend
14	dss_tptc_c1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c1_pend
13	dss_tptc_c0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c0_pend
12	dss_tptc_b1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_b1_pend
11	dss_tptc_b0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_b0_pend
10	dss_tptc_a1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_a1_pend
9	dss_tptc_a0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_a0_pend
8-5	RESERVED	R/W	0h	
4	dss_mailbox_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_mailbox_pend
3	dss_l3ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram3_pend
2	dss_l3ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram2_pend
1	dss_l3ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram1_pend
0	dss_l3ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram0_pend



#### 20.4.1.14 sec\_enable\_clr\_reg0 Register (Offset = C0h) [Reset = X]

sec\_enable\_clr\_reg0 is shown in [Table 20-39](#).

Return to the [Summary Table](#).

Interrupt Enable Clear Register 0

**Table 20-39. sec\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22	rcss_tptc_b1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_b1_pend
21	rcss_tptc_b0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_b0_pend
20	rcss_tptc_a1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_a1_pend
19	rcss_tptc_a0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_a0_pend
18	dss_tptc_c5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c5_pend
17	dss_tptc_c4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c4_pend
16	dss_tptc_c3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c3_pend
15	dss_tptc_c2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c2_pend
14	dss_tptc_c1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c1_pend
13	dss_tptc_c0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c0_pend
12	dss_tptc_b1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_b1_pend
11	dss_tptc_b0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_b0_pend
10	dss_tptc_a1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_a1_pend
9	dss_tptc_a0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_a0_pend
8-5	RESERVED	R/W	0h	
4	dss_mailbox_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_mailbox_pend
3	dss_l3ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram3_pend
2	dss_l3ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram2_pend
1	dss_l3ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram1_pend
0	dss_l3ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram0_pend

### 20.4.1.15 ded\_eoi\_reg Register (Offset = 13Ch) [Reset = X]

ded\_eoi\_reg is shown in [Table 20-40](#).

Return to the [Summary Table](#).

EOI Register

**Table 20-40. ded\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

### 20.4.1.16 ded\_status\_reg0 Register (Offset = 140h) [Reset = X]

ded\_status\_reg0 is shown in [Table 20-41](#).

Return to the [Summary Table](#).

Interrupt Status Register 0

**Table 20-41. ded\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22	rcss_tptc_b1_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_b1_pend
21	rcss_tptc_b0_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_b0_pend
20	rcss_tptc_a1_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_a1_pend
19	rcss_tptc_a0_pend	R/W1S	0h	Interrupt Pending Status for rcss_tptc_a0_pend
18	dss_tptc_c5_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c5_pend
17	dss_tptc_c4_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c4_pend
16	dss_tptc_c3_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c3_pend
15	dss_tptc_c2_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c2_pend
14	dss_tptc_c1_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c1_pend
13	dss_tptc_c0_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_c0_pend
12	dss_tptc_b1_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_b1_pend
11	dss_tptc_b0_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_b0_pend
10	dss_tptc_a1_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_a1_pend
9	dss_tptc_a0_pend	R/W1S	0h	Interrupt Pending Status for dss_tptc_a0_pend
8-5	RESERVED	R/W	0h	
4	dss_mailbox_pend	R/W1S	0h	Interrupt Pending Status for dss_mailbox_pend
3	dss_l3ram3_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram3_pend
2	dss_l3ram2_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram2_pend
1	dss_l3ram1_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram1_pend
0	dss_l3ram0_pend	R/W1S	0h	Interrupt Pending Status for dss_l3ram0_pend

### 20.4.1.17 ded\_enable\_set\_reg0 Register (Offset = 180h) [Reset = X]

ded\_enable\_set\_reg0 is shown in [Table 20-42](#).

Return to the [Summary Table](#).

Interrupt Enable Set Register 0

**Table 20-42. ded\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22	rcss_tptc_b1_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_b1_pend
21	rcss_tptc_b0_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_b0_pend
20	rcss_tptc_a1_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_a1_pend
19	rcss_tptc_a0_enable_set	R/W1S	0h	Interrupt Enable Set Register for rcss_tptc_a0_pend
18	dss_tptc_c5_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c5_pend
17	dss_tptc_c4_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c4_pend
16	dss_tptc_c3_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c3_pend
15	dss_tptc_c2_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c2_pend
14	dss_tptc_c1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c1_pend
13	dss_tptc_c0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_c0_pend
12	dss_tptc_b1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_b1_pend
11	dss_tptc_b0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_b0_pend
10	dss_tptc_a1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_a1_pend
9	dss_tptc_a0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_tptc_a0_pend
8-5	RESERVED	R/W	0h	
4	dss_mailbox_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_mailbox_pend
3	dss_l3ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram3_pend
2	dss_l3ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram2_pend
1	dss_l3ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram1_pend
0	dss_l3ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for dss_l3ram0_pend

### 20.4.1.18 ded\_enable\_clr\_reg0 Register (Offset = 1C0h) [Reset = X]

ded\_enable\_clr\_reg0 is shown in [Table 20-43](#).

Return to the [Summary Table](#).

Interrupt Enable Clear Register 0

**Table 20-43. ded\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	RESERVED	R/W	X	
22	rcss_tptc_b1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_b1_pend
21	rcss_tptc_b0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_b0_pend
20	rcss_tptc_a1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_a1_pend
19	rcss_tptc_a0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for rcss_tptc_a0_pend
18	dss_tptc_c5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c5_pend
17	dss_tptc_c4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c4_pend
16	dss_tptc_c3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c3_pend
15	dss_tptc_c2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c2_pend
14	dss_tptc_c1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c1_pend
13	dss_tptc_c0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_c0_pend
12	dss_tptc_b1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_b1_pend
11	dss_tptc_b0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_b0_pend
10	dss_tptc_a1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_a1_pend
9	dss_tptc_a0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_tptc_a0_pend
8-5	RESERVED	R/W	0h	
4	dss_mailbox_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_mailbox_pend
3	dss_l3ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram3_pend
2	dss_l3ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram2_pend
1	dss_l3ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram1_pend
0	dss_l3ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for dss_l3ram0_pend

### 20.4.1.19 aggr\_enable\_set Register (Offset = 200h) [Reset = X]

aggr\_enable\_set is shown in [Table 20-44](#).

Return to the [Summary Table](#).

AGGR interrupt enable set Register

**Table 20-44. aggr\_enable\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1S	0h	interrupt enable set for svbus timeout errors
0	parity	R/W1S	0h	interrupt enable set for parity errors

### 20.4.1.20 aggr\_enable\_clr Register (Offset = 204h) [Reset = X]

aggr\_enable\_clr is shown in [Table 20-45](#).

Return to the [Summary Table](#).

AGGR interrupt enable clear Register

**Table 20-45. aggr\_enable\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1C	0h	interrupt enable clear for svbus timeout errors
0	parity	R/W1C	0h	interrupt enable clear for parity errors

### 20.4.1.21 aggr\_status\_set Register (Offset = 208h) [Reset = X]

aggr\_status\_set is shown in [Table 20-46](#).

Return to the [Summary Table](#).

AGGR interrupt status set Register

**Table 20-46. aggr\_status\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wincr	0h	interrupt status set for svbus timeout errors
1-0	parity	R/Wincr	0h	interrupt status set for parity errors



### 20.4.1.22 aggr\_status\_clr Register (Offset = 20Ch) [Reset = X]

aggr\_status\_clr is shown in [Table 20-47](#).

Return to the [Summary Table](#).

AGGR interrupt status clear Register

**Table 20-47. aggr\_status\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wdecr	0h	interrupt status clear for svbus timeout errors
1-0	parity	R/Wdecr	0h	interrupt status clear for parity errors

## 20.4.2 MSS\_ECC\_AGGA Registers

Table 20-48 lists the MSS\_ECC\_AGGA registers. All register offset addresses not listed in Table 20-48 should be considered as reserved locations and the register contents should not be modified.

**Table 20-48. MSS\_ECC\_AGGA Registers**

Offset	Acronym	Register Name	Section
0h	rev	Aggregator Revision Register	<a href="#">Section 20.4.2.1</a>
8h	vector	ECC Vector Register	<a href="#">Section 20.4.2.2</a>
Ch	stat	Misc Status	<a href="#">Section 20.4.2.3</a>
10h	wrap_rev	ECC Wrapper Revision Register	<a href="#">Section 20.4.2.4</a>
14h	ctrl	ECC Control	<a href="#">Section 20.4.2.5</a>
18h	err_ctrl1	ECC Error Control1 Register	<a href="#">Section 20.4.2.6</a>
1Ch	err_ctrl2	ECC Error Control2 Register	<a href="#">Section 20.4.2.7</a>
20h	err_stat1	ECC Error Status1 Register	<a href="#">Section 20.4.2.8</a>
24h	err_stat2	ECC Error Status2 Register	<a href="#">Section 20.4.2.9</a>
28h	err_stat3	ECC Error Status3 Register	<a href="#">Section 20.4.2.10</a>
3Ch	sec_eoi_reg	EOI Register	<a href="#">Section 20.4.2.11</a>
40h	sec_status_reg0	Interrupt Status Register 0	<a href="#">Section 20.4.2.12</a>
80h	sec_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Section 20.4.2.13</a>
C0h	sec_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Section 20.4.2.14</a>
13Ch	ded_eoi_reg	EOI Register	<a href="#">Section 20.4.2.15</a>
140h	ded_status_reg0	Interrupt Status Register 0	<a href="#">Section 20.4.2.16</a>
180h	ded_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Section 20.4.2.17</a>
1C0h	ded_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Section 20.4.2.18</a>
200h	aggr_enable_set	AGGR interrupt enable set Register	<a href="#">Section 20.4.2.19</a>
204h	aggr_enable_clr	AGGR interrupt enable clear Register	<a href="#">Section 20.4.2.20</a>
208h	aggr_status_set	AGGR interrupt status set Register	<a href="#">Section 20.4.2.21</a>
20Ch	aggr_status_clr	AGGR interrupt status clear Register	<a href="#">Section 20.4.2.22</a>

### 20.4.2.1 rev Register (Offset = 0h) [reset = 66A0C200h]

rev is shown in [Figure 20-21](#) and described in [Table 20-49](#).

Return to the [Table 20-48](#).

Revision parameters

**Figure 20-21. rev Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu			module_id										
R-1h		R-2h			R-6A0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-18h					R-2h			R-0h			R-0h				

**Table 20-49. rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A0h	Module ID
15-11	revrtl	R	18h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	0h	Minor version

### 20.4.2.2 vector Register (Offset = 8h) [reset = X]

vector is shown in [Figure 20-22](#) and described in [Table 20-50](#).

Return to the [Table 20-48](#).

ECC Vector Register

**Figure 20-22. vector Register**

31	30	29	28	27	26	25	24
RESERVED							rd_svbus_done
R/W-X							R-0h
23	22	21	20	19	18	17	16
rd_svbus_address							
R/W-0h							
15	14	13	12	11	10	9	8
rd_svbus	RESERVED					ecc_vector	
R/W1S-0h	R/W-X					R/W-0h	
7	6	5	4	3	2	1	0
ecc_vector							
R/W-0h							

**Table 20-50. vector Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	X	
24	rd_svbus_done	R	0h	Status to indicate if read on serial VBUS is complete
23-16	rd_svbus_address	R/W	0h	Read address
15	rd_svbus	R/W1S	0h	Write 1 to trigger a read on the serial VBUS
14-11	RESERVED	R/W	X	
10-0	ecc_vector	R/W	0h	Value written to select the corresponding ECC RAM for control or status

### 20.4.2.3 stat Register (Offset = Ch) [reset = X]

stat is shown in [Figure 20-23](#) and described in [Table 20-51](#).

Return to the [Table 20-48](#).

Misc Status

**Figure 20-23. stat Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											num_rams																				
R-X											R-1Ch																				

**Table 20-51. stat Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	X	
10-0	num_rams	R	1Ch	Indicates the number of RAMS serviced by the ECC aggregator

#### 20.4.2.4 wrap\_rev Register (Offset = 10h) [reset = 66A40202h]

wrap\_rev is shown in [Figure 20-24](#) and described in [Table 20-52](#).

Return to the [Table 20-48](#).

Revision parameters

**Figure 20-24. wrap\_rev Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu		module_id											
R-1h		R-2h		R-6A4h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-0h					R-2h			R-0h		R-2h					

**Table 20-52. wrap\_rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A4h	Module ID
15-11	revrtl	R	0h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	2h	Minor version

### 20.4.2.5 ctrl Register (Offset = 14h) [reset = X]

ctrl is shown in [Figure 20-25](#) and described in [Table 20-53](#).

Return to the [Table 20-48](#).

ECC Control Register

**Figure 20-25. ctrl Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							check_svbus_timeout
R/W-X							R/W-1h
7	6	5	4	3	2	1	0
check_parity	error_once	force_n_row	force_ded	force_sec	enable_rmw	ecc_check	ecc_enable
R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

**Table 20-53. ctrl Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	check_svbus_timeout	R/W	1h	check for svbus timeout errors
7	check_parity	R/W	1h	check for parity errors
6	error_once	R/W	0h	Force Error only once
5	force_n_row	R/W	0h	Force Error on any RAM read
4	force_ded	R/W	0h	Force Double Bit Error
3	force_sec	R/W	0h	Force Single Bit Error
2	enable_rmw	R/W	1h	Enable rmw
1	ecc_check	R/W	1h	Enable ECC check
0	ecc_enable	R/W	1h	Enable ECC

### 20.4.2.6 err\_ctrl1 Register (Offset = 18h) [reset = 0h]

err\_ctrl1 is shown in [Figure 20-26](#) and described in [Table 20-54](#).

Return to the [Table 20-48](#).

ECC Error Control1 Register

**Figure 20-26. err\_ctrl1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_row																															
R/W-0h																															

**Table 20-54. err\_ctrl1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R/W	0h	Row address where single or double-bit error needs to be applied. This is ignored if force_n_row is set



### 20.4.2.7 err\_ctrl2 Register (Offset = 1Ch) [reset = 0h]

err\_ctrl2 is shown in [Figure 20-27](#) and described in [Table 20-55](#).

Return to the [Table 20-48](#).

ECC Error Control2 Register

**Figure 20-27. err\_ctrl2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_bit2																ecc_bit1															
R/W-0h																R/W-0h															

**Table 20-55. err\_ctrl2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit2	R/W	0h	Data bit that needs to be flipped if double bit error needs to be forced
15-0	ecc_bit1	R/W	0h	Data bit that needs to be flipped when force_sec is set

### 20.4.2.8 err\_stat1 Register (Offset = 20h) [reset = 0h]

err\_stat1 is shown in [Figure 20-28](#) and described in [Table 20-56](#).

Return to the [Table 20-48](#).

ECC Error Status1 Register

**Figure 20-28. err\_stat1 Register**

31	30	29	28	27	26	25	24
ecc_bit1							
R-0h							
23	22	21	20	19	18	17	16
ecc_bit1							
R-0h							
15	14	13	12	11	10	9	8
clr_ctrl_reg_err	clr_parity_err		clr_ecc_other	clr_ecc_ded		clr_ecc_sec	
R/W1C-0h	R/Wdecr-0h		R/W1C-0h	R/Wdecr-0h		R/Wdecr-0h	
7	6	5	4	3	2	1	0
ctr_reg_err	parity_err		ecc_other	ecc_ded		ecc_sec	
R/W1S-0h	R/W1S-0h		R/W1S-0h	R/Wincr-0h		R/Wincr-0h	

**Table 20-56. err\_stat1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit1	R	0h	Data bit that corresponds to the single-bit error
15	clr_ctrl_reg_err	R/W1C	0h	Clear control reg error Error Status, you must also re write the control register itself to clear this
14-13	clr_parity_err	R/Wdecr	0h	Clear parity Error Status
12	clr_ecc_other	R/W1C	0h	Clear other Error Status
11-10	clr_ecc_ded	R/Wdecr	0h	Clear Double Bit Error Status
9-8	clr_ecc_sec	R/Wdecr	0h	Clear Single Bit Error Status
7	ctr_reg_err	R/W1S	0h	control register error pending, Level interrupt
6-5	parity_err	R/W1S	0h	Level parity error Error Status
4	ecc_other	R/W1S	0h	successive single-bit errors have occurred while a writeback is still pending, Level interrupt
3-2	ecc_ded	R/Wincr	0h	Level Double Bit Error Status
1-0	ecc_sec	R/Wincr	0h	Level Single Bit Error Status

#### 20.4.2.9 err\_stat2 Register (Offset = 24h) [reset = 0h]

err\_stat2 is shown in [Figure 20-29](#) and described in [Table 20-57](#).

Return to the [Table 20-48](#).

ECC Error Status2 Register

**Figure 20-29. err\_stat2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_row																															
R-0h																															

**Table 20-57. err\_stat2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R	0h	Row address where the single or double-bit error has occurred

### 20.4.2.10 err\_stat3 Register (Offset = 28h) [reset = X]

err\_stat3 is shown in [Figure 20-30](#) and described in [Table 20-58](#).

Return to the [Table 20-48](#).

ECC Error Status3 Register

**Figure 20-30. err\_stat3 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED						clr_svbus_timeo ut_err	RESERVED
R/W-X						R/W1C-0h	R/W-X
7	6	5	4	3	2	1	0
RESERVED						svbus_timeout_ err	wb_pend
R/W-X						R/W1S-0h	R-0h

**Table 20-58. err\_stat3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	
9	clr_svbus_timeout_err	R/W1C	0h	Clear svbus timeout Error Status
8-2	RESERVED	R/W	X	
1	svbus_timeout_err	R/W1S	0h	Level svbus timeout error Error Status
0	wb_pend	R	0h	delayed write back pending Status

### 20.4.2.11 sec\_eoi\_reg Register (Offset = 3Ch) [reset = X]

sec\_eoi\_reg is shown in [Figure 20-31](#) and described in [Table 20-59](#).

Return to the [Table 20-48](#).

EOI Register

**Figure 20-31. sec\_eoi\_reg Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-X							R/W1S-0h

**Table 20-59. sec\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

### 20.4.2.12 sec\_status\_reg0 Register (Offset = 40h) [reset = X]

sec\_status\_reg0 is shown in [Figure 20-32](#) and described in [Table 20-60](#).

Return to the [Table 20-48](#).

Interrupt Status Register 0

**Figure 20-32. sec\_status\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu0_ks_vim_r amecc_pend	b1tcm0_bank1_ pend	b1tcm0_bank0_ pend	b0tcm0_bank1_ pend
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm0_bank0_ pend	atcm0_bank1_p end	atcm0_bank0_p end	cpu0_ddata_ra m7_pend	cpu0_ddata_ra m6_pend	cpu0_ddata_ra m5_pend	cpu0_ddata_ra m4_pend	cpu0_ddata_ra m3_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu0_ddata_ra m2_pend	cpu0_ddata_ra m1_pend	cpu0_ddata_ra m0_pend	cpu0_ddirty_ra m_pend	cpu0_dtag_ram 3_pend	cpu0_dtag_ram 2_pend	cpu0_dtag_ram 1_pend	cpu0_dtag_ram 0_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu0_idata_ban k3_pend	cpu0_idata_ban k2_pend	cpu0_idata_ban k1_pend	cpu0_idata_ban k0_pend	cpu0_itag_ram3 _pend	cpu0_itag_ram2 _pend	cpu0_itag_ram1 _pend	cpu0_itag_ram0 _pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-60. sec\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu0_ks_vim_ramecc_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ks_vim_ramecc_pend
26	b1tcm0_bank1_pend	R/W1S	0h	Interrupt Pending Status for b1tcm0_bank1_pend
25	b1tcm0_bank0_pend	R/W1S	0h	Interrupt Pending Status for b1tcm0_bank0_pend
24	b0tcm0_bank1_pend	R/W1S	0h	Interrupt Pending Status for b0tcm0_bank1_pend
23	b0tcm0_bank0_pend	R/W1S	0h	Interrupt Pending Status for b0tcm0_bank0_pend
22	atcm0_bank1_pend	R/W1S	0h	Interrupt Pending Status for atcm0_bank1_pend
21	atcm0_bank0_pend	R/W1S	0h	Interrupt Pending Status for atcm0_bank0_pend
20	cpu0_ddata_ram7_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram7_pend
19	cpu0_ddata_ram6_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram6_pend
18	cpu0_ddata_ram5_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram5_pend
17	cpu0_ddata_ram4_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram4_pend
16	cpu0_ddata_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram3_pend
15	cpu0_ddata_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram2_pend
14	cpu0_ddata_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram1_pend
13	cpu0_ddata_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram0_pend
12	cpu0_ddirty_ram_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddirty_ram_pend
11	cpu0_dtag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram3_pend
10	cpu0_dtag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram2_pend

**Table 20-60. sec\_status\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9	cpu0_dtag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram1_pend
8	cpu0_dtag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram0_pend
7	cpu0_idata_bank3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank3_pend
6	cpu0_idata_bank2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank2_pend
5	cpu0_idata_bank1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank1_pend
4	cpu0_idata_bank0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank0_pend
3	cpu0_itag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram3_pend
2	cpu0_itag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram2_pend
1	cpu0_itag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram1_pend
0	cpu0_itag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram0_pend

### 20.4.2.13 sec\_enable\_set\_reg0 Register (Offset = 80h) [reset = X]

sec\_enable\_set\_reg0 is shown in [Figure 20-33](#) and described in [Table 20-61](#).

Return to the [Table 20-48](#).

Interrupt Enable Set Register 0

**Figure 20-33. sec\_enable\_set\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu0_ks_vim_ramecc_enable_set	b1tcm0_bank1_enable_set	b1tcm0_bank0_enable_set	b0tcm0_bank1_enable_set
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm0_bank0_enable_set	atcm0_bank1_enable_set	atcm0_bank0_enable_set	cpu0_ddata_ram7_enable_set	cpu0_ddata_ram6_enable_set	cpu0_ddata_ram5_enable_set	cpu0_ddata_ram4_enable_set	cpu0_ddata_ram3_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu0_ddata_ram2_enable_set	cpu0_ddata_ram1_enable_set	cpu0_ddata_ram0_enable_set	cpu0_ddirty_ram_enable_set	cpu0_dtag_ram3_enable_set	cpu0_dtag_ram2_enable_set	cpu0_dtag_ram1_enable_set	cpu0_dtag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu0_idata_bank3_enable_set	cpu0_idata_bank2_enable_set	cpu0_idata_bank1_enable_set	cpu0_idata_bank0_enable_set	cpu0_itag_ram3_enable_set	cpu0_itag_ram2_enable_set	cpu0_itag_ram1_enable_set	cpu0_itag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-61. sec\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu0_ks_vim_ramecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ks_vim_ramecc_pend
26	b1tcm0_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm0_bank1_pend
25	b1tcm0_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm0_bank0_pend
24	b0tcm0_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm0_bank1_pend
23	b0tcm0_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm0_bank0_pend
22	atcm0_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm0_bank1_pend
21	atcm0_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm0_bank0_pend
20	cpu0_ddata_ram7_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram7_pend
19	cpu0_ddata_ram6_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram6_pend
18	cpu0_ddata_ram5_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram5_pend
17	cpu0_ddata_ram4_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram4_pend
16	cpu0_ddata_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram3_pend
15	cpu0_ddata_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram2_pend



**Table 20-61. sec\_enable\_set\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	cpu0_ddata_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram1_pend
13	cpu0_ddata_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram0_pend
12	cpu0_ddirty_ram_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddirty_ram_pend
11	cpu0_dtag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram3_pend
10	cpu0_dtag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram2_pend
9	cpu0_dtag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram1_pend
8	cpu0_dtag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram0_pend
7	cpu0_idata_bank3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank3_pend
6	cpu0_idata_bank2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank2_pend
5	cpu0_idata_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank1_pend
4	cpu0_idata_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank0_pend
3	cpu0_itag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram3_pend
2	cpu0_itag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram2_pend
1	cpu0_itag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram1_pend
0	cpu0_itag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram0_pend

### 20.4.2.14 sec\_enable\_clr\_reg0 Register (Offset = C0h) [reset = X]

sec\_enable\_clr\_reg0 is shown in [Figure 20-34](#) and described in [Table 20-62](#).

Return to the [Table 20-48](#).

Interrupt Enable Clear Register 0

**Figure 20-34. sec\_enable\_clr\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu0_ks_vim_ramecc_enable_clr	b1tcm0_bank1_enable_clr	b1tcm0_bank0_enable_clr	b0tcm0_bank1_enable_clr
R/W-X				R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
b0tcm0_bank0_enable_clr	atcm0_bank1_enable_clr	atcm0_bank0_enable_clr	cpu0_ddata_ram7_enable_clr	cpu0_ddata_ram6_enable_clr	cpu0_ddata_ram5_enable_clr	cpu0_ddata_ram4_enable_clr	cpu0_ddata_ram3_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
cpu0_ddata_ram2_enable_clr	cpu0_ddata_ram1_enable_clr	cpu0_ddata_ram0_enable_clr	cpu0_ddirty_ram_enable_clr	cpu0_dtag_ram3_enable_clr	cpu0_dtag_ram2_enable_clr	cpu0_dtag_ram1_enable_clr	cpu0_dtag_ram0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
cpu0_idata_bank3_enable_clr	cpu0_idata_bank2_enable_clr	cpu0_idata_bank1_enable_clr	cpu0_idata_bank0_enable_clr	cpu0_itag_ram3_enable_clr	cpu0_itag_ram2_enable_clr	cpu0_itag_ram1_enable_clr	cpu0_itag_ram0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**Table 20-62. sec\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu0_ks_vim_ramecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ks_vim_ramecc_pend
26	b1tcm0_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm0_bank1_pend
25	b1tcm0_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm0_bank0_pend
24	b0tcm0_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm0_bank1_pend
23	b0tcm0_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm0_bank0_pend
22	atcm0_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm0_bank1_pend
21	atcm0_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm0_bank0_pend
20	cpu0_ddata_ram7_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram7_pend
19	cpu0_ddata_ram6_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram6_pend
18	cpu0_ddata_ram5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram5_pend
17	cpu0_ddata_ram4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram4_pend
16	cpu0_ddata_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram3_pend
15	cpu0_ddata_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram2_pend
14	cpu0_ddata_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram1_pend

**Table 20-62. sec\_enable\_clr\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
13	cpu0_ddata_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram0_pend
12	cpu0_ddirty_ram_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddirty_ram_pend
11	cpu0_dtag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram3_pend
10	cpu0_dtag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram2_pend
9	cpu0_dtag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram1_pend
8	cpu0_dtag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram0_pend
7	cpu0_idata_bank3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank3_pend
6	cpu0_idata_bank2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank2_pend
5	cpu0_idata_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank1_pend
4	cpu0_idata_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank0_pend
3	cpu0_itag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram3_pend
2	cpu0_itag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram2_pend
1	cpu0_itag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram1_pend
0	cpu0_itag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram0_pend

### 20.4.2.15 ded\_eoi\_reg Register (Offset = 13Ch) [reset = X]

ded\_eoi\_reg is shown in [Figure 20-35](#) and described in [Table 20-63](#).

Return to the [Table 20-48](#).

EOI Register

**Figure 20-35. ded\_eoi\_reg Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-X							R/W1S-0h

**Table 20-63. ded\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

#### 20.4.2.16 ded\_status\_reg0 Register (Offset = 140h) [reset = X]

ded\_status\_reg0 is shown in [Figure 20-36](#) and described in [Table 20-64](#).

Return to the [Table 20-48](#).

Interrupt Status Register 0

**Figure 20-36. ded\_status\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu0_ks_vim_r amecc_pend	b1tcm0_bank1_ pend	b1tcm0_bank0_ pend	b0tcm0_bank1_ pend
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm0_bank0_ pend	atcm0_bank1_p end	atcm0_bank0_p end	cpu0_ddata_ra m7_pend	cpu0_ddata_ra m6_pend	cpu0_ddata_ra m5_pend	cpu0_ddata_ra m4_pend	cpu0_ddata_ra m3_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu0_ddata_ra m2_pend	cpu0_ddata_ra m1_pend	cpu0_ddata_ra m0_pend	cpu0_ddirty_ra m_pend	cpu0_dtag_ram 3_pend	cpu0_dtag_ram 2_pend	cpu0_dtag_ram 1_pend	cpu0_dtag_ram 0_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu0_idata_ban k3_pend	cpu0_idata_ban k2_pend	cpu0_idata_ban k1_pend	cpu0_idata_ban k0_pend	cpu0_itag_ram3 _pend	cpu0_itag_ram2 _pend	cpu0_itag_ram1 _pend	cpu0_itag_ram0 _pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-64. ded\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu0_ks_vim_ramecc_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ks_vim_ramecc_pend
26	b1tcm0_bank1_pend	R/W1S	0h	Interrupt Pending Status for b1tcm0_bank1_pend
25	b1tcm0_bank0_pend	R/W1S	0h	Interrupt Pending Status for b1tcm0_bank0_pend
24	b0tcm0_bank1_pend	R/W1S	0h	Interrupt Pending Status for b0tcm0_bank1_pend
23	b0tcm0_bank0_pend	R/W1S	0h	Interrupt Pending Status for b0tcm0_bank0_pend
22	atcm0_bank1_pend	R/W1S	0h	Interrupt Pending Status for atcm0_bank1_pend
21	atcm0_bank0_pend	R/W1S	0h	Interrupt Pending Status for atcm0_bank0_pend
20	cpu0_ddata_ram7_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram7_pend
19	cpu0_ddata_ram6_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram6_pend
18	cpu0_ddata_ram5_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram5_pend
17	cpu0_ddata_ram4_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram4_pend
16	cpu0_ddata_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram3_pend
15	cpu0_ddata_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram2_pend
14	cpu0_ddata_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram1_pend
13	cpu0_ddata_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddata_ram0_pend
12	cpu0_ddirty_ram_pend	R/W1S	0h	Interrupt Pending Status for cpu0_ddirty_ram_pend
11	cpu0_dtag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram3_pend
10	cpu0_dtag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram2_pend

**Table 20-64. ded\_status\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9	cpu0_dtag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram1_pend
8	cpu0_dtag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_dtag_ram0_pend
7	cpu0_idata_bank3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank3_pend
6	cpu0_idata_bank2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank2_pend
5	cpu0_idata_bank1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank1_pend
4	cpu0_idata_bank0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_idata_bank0_pend
3	cpu0_itag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram3_pend
2	cpu0_itag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram2_pend
1	cpu0_itag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram1_pend
0	cpu0_itag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu0_itag_ram0_pend

### 20.4.2.17 ded\_enable\_set\_reg0 Register (Offset = 180h) [reset = X]

ded\_enable\_set\_reg0 is shown in [Figure 20-37](#) and described in [Table 20-65](#).

Return to the [Table 20-48](#).

Interrupt Enable Set Register 0

**Figure 20-37. ded\_enable\_set\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu0_ks_vim_ramecc_enable_set	b1tcm0_bank1_enable_set	b1tcm0_bank0_enable_set	b0tcm0_bank1_enable_set
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm0_bank0_enable_set	atcm0_bank1_enable_set	atcm0_bank0_enable_set	cpu0_ddata_ram7_enable_set	cpu0_ddata_ram6_enable_set	cpu0_ddata_ram5_enable_set	cpu0_ddata_ram4_enable_set	cpu0_ddata_ram3_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu0_ddata_ram2_enable_set	cpu0_ddata_ram1_enable_set	cpu0_ddata_ram0_enable_set	cpu0_ddirty_ram_enable_set	cpu0_dtag_ram3_enable_set	cpu0_dtag_ram2_enable_set	cpu0_dtag_ram1_enable_set	cpu0_dtag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu0_idata_bank3_enable_set	cpu0_idata_bank2_enable_set	cpu0_idata_bank1_enable_set	cpu0_idata_bank0_enable_set	cpu0_itag_ram3_enable_set	cpu0_itag_ram2_enable_set	cpu0_itag_ram1_enable_set	cpu0_itag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-65. ded\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu0_ks_vim_ramecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ks_vim_ramecc_pend
26	b1tcm0_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm0_bank1_pend
25	b1tcm0_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm0_bank0_pend
24	b0tcm0_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm0_bank1_pend
23	b0tcm0_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm0_bank0_pend
22	atcm0_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm0_bank1_pend
21	atcm0_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm0_bank0_pend
20	cpu0_ddata_ram7_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram7_pend
19	cpu0_ddata_ram6_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram6_pend
18	cpu0_ddata_ram5_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram5_pend
17	cpu0_ddata_ram4_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram4_pend
16	cpu0_ddata_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram3_pend
15	cpu0_ddata_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram2_pend

**Table 20-65. ded\_enable\_set\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	cpu0_ddata_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram1_pend
13	cpu0_ddata_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddata_ram0_pend
12	cpu0_ddirty_ram_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_ddirty_ram_pend
11	cpu0_dtag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram3_pend
10	cpu0_dtag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram2_pend
9	cpu0_dtag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram1_pend
8	cpu0_dtag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_dtag_ram0_pend
7	cpu0_idata_bank3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank3_pend
6	cpu0_idata_bank2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank2_pend
5	cpu0_idata_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank1_pend
4	cpu0_idata_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_idata_bank0_pend
3	cpu0_itag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram3_pend
2	cpu0_itag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram2_pend
1	cpu0_itag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram1_pend
0	cpu0_itag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu0_itag_ram0_pend



### 20.4.2.18 ded\_enable\_clr\_reg0 Register (Offset = 1C0h) [reset = X]

ded\_enable\_clr\_reg0 is shown in [Figure 20-38](#) and described in [Table 20-66](#).

Return to the [Table 20-48](#).

Interrupt Enable Clear Register 0

**Figure 20-38. ded\_enable\_clr\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu0_ks_vim_ramecc_enable_clr	b1tcm0_bank1_enable_clr	b1tcm0_bank0_enable_clr	b0tcm0_bank1_enable_clr
R/W-X				R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
b0tcm0_bank0_enable_clr	atcm0_bank1_enable_clr	atcm0_bank0_enable_clr	cpu0_ddata_ram7_enable_clr	cpu0_ddata_ram6_enable_clr	cpu0_ddata_ram5_enable_clr	cpu0_ddata_ram4_enable_clr	cpu0_ddata_ram3_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
cpu0_ddata_ram2_enable_clr	cpu0_ddata_ram1_enable_clr	cpu0_ddata_ram0_enable_clr	cpu0_ddirty_ram_enable_clr	cpu0_dtag_ram3_enable_clr	cpu0_dtag_ram2_enable_clr	cpu0_dtag_ram1_enable_clr	cpu0_dtag_ram0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
cpu0_idata_bank3_enable_clr	cpu0_idata_bank2_enable_clr	cpu0_idata_bank1_enable_clr	cpu0_idata_bank0_enable_clr	cpu0_itag_ram3_enable_clr	cpu0_itag_ram2_enable_clr	cpu0_itag_ram1_enable_clr	cpu0_itag_ram0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**Table 20-66. ded\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu0_ks_vim_ramecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ks_vim_ramecc_pend
26	b1tcm0_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm0_bank1_pend
25	b1tcm0_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm0_bank0_pend
24	b0tcm0_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm0_bank1_pend
23	b0tcm0_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm0_bank0_pend
22	atcm0_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm0_bank1_pend
21	atcm0_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm0_bank0_pend
20	cpu0_ddata_ram7_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram7_pend
19	cpu0_ddata_ram6_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram6_pend
18	cpu0_ddata_ram5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram5_pend
17	cpu0_ddata_ram4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram4_pend
16	cpu0_ddata_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram3_pend
15	cpu0_ddata_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram2_pend
14	cpu0_ddata_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram1_pend

**Table 20-66. ded\_enable\_clr\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
13	cpu0_ddata_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddata_ram0_pend
12	cpu0_ddirty_ram_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_ddirty_ram_pend
11	cpu0_dtag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram3_pend
10	cpu0_dtag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram2_pend
9	cpu0_dtag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram1_pend
8	cpu0_dtag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_dtag_ram0_pend
7	cpu0_idata_bank3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank3_pend
6	cpu0_idata_bank2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank2_pend
5	cpu0_idata_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank1_pend
4	cpu0_idata_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_idata_bank0_pend
3	cpu0_itag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram3_pend
2	cpu0_itag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram2_pend
1	cpu0_itag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram1_pend
0	cpu0_itag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu0_itag_ram0_pend

### 20.4.2.19 aggr\_enable\_set Register (Offset = 200h) [reset = X]

aggr\_enable\_set is shown in [Figure 20-39](#) and described in [Table 20-67](#).

Return to the [Table 20-48](#).

AGGR interrupt enable set Register

**Figure 20-39. aggr\_enable\_set Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-X						R/W1S-0h	R/W1S-0h

**Table 20-67. aggr\_enable\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1S	0h	interrupt enable set for svbus timeout errors
0	parity	R/W1S	0h	interrupt enable set for parity errors

### 20.4.2.20 aggr\_enable\_clr Register (Offset = 204h) [reset = X]

aggr\_enable\_clr is shown in [Figure 20-40](#) and described in [Table 20-68](#).

Return to the [Table 20-48](#).

AGGR interrupt enable clear Register

**Figure 20-40. aggr\_enable\_clr Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-X						R/W1C-0h	R/W1C-0h

**Table 20-68. aggr\_enable\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1C	0h	interrupt enable clear for svbus timeout errors
0	parity	R/W1C	0h	interrupt enable clear for parity errors

### 20.4.2.21 aggr\_status\_set Register (Offset = 208h) [reset = X]

aggr\_status\_set is shown in [Figure 20-41](#) and described in [Table 20-69](#).

Return to the [Table 20-48](#).

AGGR interrupt status set Register

**Figure 20-41. aggr\_status\_set Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				timeout		parity	
R/W-X				R/Wincr-0h		R/Wincr-0h	

**Table 20-69. aggr\_status\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wincr	0h	interrupt status set for svbus timeout errors
1-0	parity	R/Wincr	0h	interrupt status set for parity errors

### 20.4.2.22 aggr\_status\_clr Register (Offset = 20Ch) [reset = X]

aggr\_status\_clr is shown in [Figure 20-42](#) and described in [Table 20-70](#).

Return to the [Table 20-48](#).

AGGR interrupt status clear Register

**Figure 20-42. aggr\_status\_clr Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				timeout		parity	
R/W-X				R/Wdecr-0h		R/Wdecr-0h	

**Table 20-70. aggr\_status\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wdecr	0h	interrupt status clear for svbus timeout errors
1-0	parity	R/Wdecr	0h	interrupt status clear for parity errors

### 20.4.3 MSS\_ECC\_AGGB Registers

Table 20-71 lists the MSS\_ECC\_AGGB registers. All register offset addresses not listed in Table 20-71 should be considered as reserved locations and the register contents should not be modified.

**Table 20-71. MSS\_ECC\_AGGB Registers**

Offset	Acronym	Register Name	Section
0h	rev	Aggregator Revision Register	<a href="#">Section 20.4.3.1</a>
8h	vector	ECC Vector Register	<a href="#">Section 20.4.3.2</a>
Ch	stat	Misc Status	<a href="#">Section 20.4.3.3</a>
10h	wrap_rev	ECC Wrapper Revision Register	<a href="#">Section 20.4.3.4</a>
14h	ctrl	ECC Control	<a href="#">Section 20.4.3.5</a>
18h	err_ctrl1	ECC Error Control1 Register	<a href="#">Section 20.4.3.6</a>
1Ch	err_ctrl2	ECC Error Control2 Register	<a href="#">Section 20.4.3.7</a>
20h	err_stat1	ECC Error Status1 Register	<a href="#">Section 20.4.3.8</a>
24h	err_stat2	ECC Error Status2 Register	<a href="#">Section 20.4.3.9</a>
28h	err_stat3	ECC Error Status3 Register	<a href="#">Section 20.4.3.10</a>
3Ch	sec_eoi_reg	EOI Register	<a href="#">Section 20.4.3.11</a>
40h	sec_status_reg0	Interrupt Status Register 0	<a href="#">Section 20.4.3.12</a>
80h	sec_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Section 20.4.3.13</a>
C0h	sec_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Section 20.4.3.14</a>
13Ch	ded_eoi_reg	EOI Register	<a href="#">Section 20.4.3.15</a>
140h	ded_status_reg0	Interrupt Status Register 0	<a href="#">Section 20.4.3.16</a>
180h	ded_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Section 20.4.3.17</a>
1C0h	ded_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Section 20.4.3.18</a>
200h	aggr_enable_set	AGGR interrupt enable set Register	<a href="#">Section 20.4.3.19</a>
204h	aggr_enable_clr	AGGR interrupt enable clear Register	<a href="#">Section 20.4.3.20</a>
208h	aggr_status_set	AGGR interrupt status set Register	<a href="#">Section 20.4.3.21</a>
20Ch	aggr_status_clr	AGGR interrupt status clear Register	<a href="#">Section 20.4.3.22</a>

### 20.4.3.1 rev Register (Offset = 0h) [reset = 66A0C200h]

rev is shown in [Figure 20-43](#) and described in [Table 20-72](#).

Return to the [Table 20-71](#).

Revision parameters

**Figure 20-43. rev Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu			module_id										
R-1h		R-2h			R-6A0h										
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-18h					R-2h			R-0h			R-0h				

**Table 20-72. rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A0h	Module ID
15-11	revrtl	R	18h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	0h	Minor version



### 20.4.3.2 vector Register (Offset = 8h) [reset = X]

vector is shown in [Figure 20-44](#) and described in [Table 20-73](#).

Return to the [Table 20-71](#).

ECC Vector Register

**Figure 20-44. vector Register**

31	30	29	28	27	26	25	24
RESERVED							rd_svbus_done
R/W-X							R-0h
23	22	21	20	19	18	17	16
rd_svbus_address							
R/W-0h							
15	14	13	12	11	10	9	8
rd_svbus	RESERVED					ecc_vector	
R/W1S-0h	R/W-X					R/W-0h	
7	6	5	4	3	2	1	0
ecc_vector							
R/W-0h							

**Table 20-73. vector Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	X	
24	rd_svbus_done	R	0h	Status to indicate if read on serial VBUS is complete
23-16	rd_svbus_address	R/W	0h	Read address
15	rd_svbus	R/W1S	0h	Write 1 to trigger a read on the serial VBUS
14-11	RESERVED	R/W	X	
10-0	ecc_vector	R/W	0h	Value written to select the corresponding ECC RAM for control or status

### 20.4.3.3 stat Register (Offset = Ch) [reset = X]

stat is shown in [Figure 20-45](#) and described in [Table 20-74](#).

Return to the [Table 20-71](#).

Misc Status

**Figure 20-45. stat Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0						
RESERVED																					num_rams																
R-X																					R-1Ch																

**Table 20-74. stat Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	X	
10-0	num_rams	R	1Ch	Indicates the number of RAMS serviced by the ECC aggregator

#### 20.4.3.4 wrap\_rev Register (Offset = 10h) [reset = 66A40202h]

wrap\_rev is shown in [Figure 20-46](#) and described in [Table 20-75](#).

Return to the [Table 20-71](#).

Revision parameters

**Figure 20-46. wrap\_rev Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu		module_id											
R-1h		R-2h		R-6A4h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-0h					R-2h			R-0h		R-2h					

**Table 20-75. wrap\_rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A4h	Module ID
15-11	revrtl	R	0h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	2h	Minor version

### 20.4.3.5 ctrl Register (Offset = 14h) [reset = X]

ctrl is shown in [Figure 20-47](#) and described in [Table 20-76](#).

Return to the [Table 20-71](#).

ECC Control Register

**Figure 20-47. ctrl Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							check_svbus_timeout
R/W-X							R/W-1h
7	6	5	4	3	2	1	0
check_parity	error_once	force_n_row	force_ded	force_sec	enable_rmw	ecc_check	ecc_enable
R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

**Table 20-76. ctrl Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	check_svbus_timeout	R/W	1h	check for svbus timeout errors
7	check_parity	R/W	1h	check for parity errors
6	error_once	R/W	0h	Force Error only once
5	force_n_row	R/W	0h	Force Error on any RAM read
4	force_ded	R/W	0h	Force Double Bit Error
3	force_sec	R/W	0h	Force Single Bit Error
2	enable_rmw	R/W	1h	Enable rmw
1	ecc_check	R/W	1h	Enable ECC check
0	ecc_enable	R/W	1h	Enable ECC

### 20.4.3.6 err\_ctrl1 Register (Offset = 18h) [reset = 0h]

err\_ctrl1 is shown in [Figure 20-48](#) and described in [Table 20-77](#).

Return to the [Table 20-71](#).

ECC Error Control1 Register

**Figure 20-48. err\_ctrl1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_row																															
R/W-0h																															

**Table 20-77. err\_ctrl1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R/W	0h	Row address where single or double-bit error needs to be applied. This is ignored if force_n_row is set

### 20.4.3.7 err\_ctrl2 Register (Offset = 1Ch) [reset = 0h]

err\_ctrl2 is shown in [Figure 20-49](#) and described in [Table 20-78](#).

Return to the [Table 20-71](#).

ECC Error Control2 Register

**Figure 20-49. err\_ctrl2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_bit2																ecc_bit1															
R/W-0h																R/W-0h															

**Table 20-78. err\_ctrl2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit2	R/W	0h	Data bit that needs to be flipped if double bit error needs to be forced
15-0	ecc_bit1	R/W	0h	Data bit that needs to be flipped when force_sec is set

### 20.4.3.8 err\_stat1 Register (Offset = 20h) [reset = 0h]

err\_stat1 is shown in [Figure 20-50](#) and described in [Table 20-79](#).

Return to the [Table 20-71](#).

ECC Error Status1 Register

**Figure 20-50. err\_stat1 Register**

31	30	29	28	27	26	25	24
ecc_bit1							
R-0h							
23	22	21	20	19	18	17	16
ecc_bit1							
R-0h							
15	14	13	12	11	10	9	8
clr_ctrl_reg_err	clr_parity_err		clr_ecc_other	clr_ecc_ded		clr_ecc_sec	
R/W1C-0h	R/Wdecr-0h		R/W1C-0h	R/Wdecr-0h		R/Wdecr-0h	
7	6	5	4	3	2	1	0
ctr_reg_err	parity_err		ecc_other	ecc_ded		ecc_sec	
R/W1S-0h	R/W1S-0h		R/W1S-0h	R/Wincr-0h		R/Wincr-0h	

**Table 20-79. err\_stat1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit1	R	0h	Data bit that corresponds to the single-bit error
15	clr_ctrl_reg_err	R/W1C	0h	Clear control reg error Error Status, you must also re write the control register itself to clear this
14-13	clr_parity_err	R/Wdecr	0h	Clear parity Error Status
12	clr_ecc_other	R/W1C	0h	Clear other Error Status
11-10	clr_ecc_ded	R/Wdecr	0h	Clear Double Bit Error Status
9-8	clr_ecc_sec	R/Wdecr	0h	Clear Single Bit Error Status
7	ctr_reg_err	R/W1S	0h	control register error pending, Level interrupt
6-5	parity_err	R/W1S	0h	Level parity error Error Status
4	ecc_other	R/W1S	0h	successive single-bit errors have occurred while a writeback is still pending, Level interrupt
3-2	ecc_ded	R/Wincr	0h	Level Double Bit Error Status
1-0	ecc_sec	R/Wincr	0h	Level Single Bit Error Status

### 20.4.3.9 err\_stat2 Register (Offset = 24h) [reset = 0h]

err\_stat2 is shown in [Figure 20-51](#) and described in [Table 20-80](#).

Return to the [Table 20-71](#).

ECC Error Status2 Register

**Figure 20-51. err\_stat2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_row																															
R-0h																															

**Table 20-80. err\_stat2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R	0h	Row address where the single or double-bit error has occurred



### 20.4.3.10 err\_stat3 Register (Offset = 28h) [reset = X]

err\_stat3 is shown in [Figure 20-52](#) and described in [Table 20-81](#).

Return to the [Table 20-71](#).

ECC Error Status3 Register

**Figure 20-52. err\_stat3 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED						clr_svbus_timeo ut_err	RESERVED
R/W-X						R/W1C-0h	R/W-X
7	6	5	4	3	2	1	0
RESERVED						svbus_timeout_ err	wb_pend
R/W-X						R/W1S-0h	R-0h

**Table 20-81. err\_stat3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	
9	clr_svbus_timeout_err	R/W1C	0h	Clear svbus timeout Error Status
8-2	RESERVED	R/W	X	
1	svbus_timeout_err	R/W1S	0h	Level svbus timeout error Error Status
0	wb_pend	R	0h	delayed write back pending Status

### 20.4.3.11 sec\_eoi\_reg Register (Offset = 3Ch) [reset = X]

sec\_eoi\_reg is shown in [Figure 20-53](#) and described in [Table 20-82](#).

Return to the [Table 20-71](#).

EOI Register

**Figure 20-53. sec\_eoi\_reg Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-X							R/W1S-0h

**Table 20-82. sec\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

### 20.4.3.12 sec\_status\_reg0 Register (Offset = 40h) [reset = X]

sec\_status\_reg0 is shown in [Figure 20-54](#) and described in [Table 20-83](#).

Return to the [Table 20-71](#).

Interrupt Status Register 0

**Figure 20-54. sec\_status\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu1_ks_vim_r amecc_pend	b1tcm1_bank1_ pend	b1tcm1_bank0_ pend	b0tcm1_bank1_ pend
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm1_bank0_ pend	atcm1_bank1_p end	atcm1_bank0_p end	cpu1_ddata_ra m7_pend	cpu1_ddata_ra m6_pend	cpu1_ddata_ra m5_pend	cpu1_ddata_ra m4_pend	cpu1_ddata_ra m3_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu1_ddata_ra m2_pend	cpu1_ddata_ra m1_pend	cpu1_ddata_ra m0_pend	cpu1_ddirty_ra m_pend	cpu1_dtag_ram 3_pend	cpu1_dtag_ram 2_pend	cpu1_dtag_ram 1_pend	cpu1_dtag_ram 0_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu1_idata_ban k3_pend	cpu1_idata_ban k2_pend	cpu1_idata_ban k1_pend	cpu1_idata_ban k0_pend	cpu1_itag_ram3 _pend	cpu1_itag_ram2 _pend	cpu1_itag_ram1 _pend	cpu1_itag_ram0 _pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-83. sec\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu1_ks_vim_ramecc_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ks_vim_ramecc_pend
26	b1tcm1_bank1_pend	R/W1S	0h	Interrupt Pending Status for b1tcm1_bank1_pend
25	b1tcm1_bank0_pend	R/W1S	0h	Interrupt Pending Status for b1tcm1_bank0_pend
24	b0tcm1_bank1_pend	R/W1S	0h	Interrupt Pending Status for b0tcm1_bank1_pend
23	b0tcm1_bank0_pend	R/W1S	0h	Interrupt Pending Status for b0tcm1_bank0_pend
22	atcm1_bank1_pend	R/W1S	0h	Interrupt Pending Status for atcm1_bank1_pend
21	atcm1_bank0_pend	R/W1S	0h	Interrupt Pending Status for atcm1_bank0_pend
20	cpu1_ddata_ram7_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram7_pend
19	cpu1_ddata_ram6_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram6_pend
18	cpu1_ddata_ram5_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram5_pend
17	cpu1_ddata_ram4_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram4_pend
16	cpu1_ddata_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram3_pend
15	cpu1_ddata_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram2_pend
14	cpu1_ddata_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram1_pend
13	cpu1_ddata_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram0_pend
12	cpu1_ddirty_ram_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddirty_ram_pend
11	cpu1_dtag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram3_pend
10	cpu1_dtag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram2_pend

**Table 20-83. sec\_status\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9	cpu1_dtag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram1_pend
8	cpu1_dtag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram0_pend
7	cpu1_idata_bank3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank3_pend
6	cpu1_idata_bank2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank2_pend
5	cpu1_idata_bank1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank1_pend
4	cpu1_idata_bank0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank0_pend
3	cpu1_itag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram3_pend
2	cpu1_itag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram2_pend
1	cpu1_itag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram1_pend
0	cpu1_itag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram0_pend

### 20.4.3.13 sec\_enable\_set\_reg0 Register (Offset = 80h) [reset = X]

sec\_enable\_set\_reg0 is shown in [Figure 20-55](#) and described in [Table 20-84](#).

Return to the [Table 20-71](#).

Interrupt Enable Set Register 0

**Figure 20-55. sec\_enable\_set\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu1_ks_vim_ramecc_enable_set	b1tcm1_bank1_enable_set	b1tcm1_bank0_enable_set	b0tcm1_bank1_enable_set
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm1_bank0_enable_set	atcm1_bank1_enable_set	atcm1_bank0_enable_set	cpu1_ddata_ram7_enable_set	cpu1_ddata_ram6_enable_set	cpu1_ddata_ram5_enable_set	cpu1_ddata_ram4_enable_set	cpu1_ddata_ram3_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu1_ddata_ram2_enable_set	cpu1_ddata_ram1_enable_set	cpu1_ddata_ram0_enable_set	cpu1_ddirty_ram_enable_set	cpu1_dtag_ram3_enable_set	cpu1_dtag_ram2_enable_set	cpu1_dtag_ram1_enable_set	cpu1_dtag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu1_idata_bank3_enable_set	cpu1_idata_bank2_enable_set	cpu1_idata_bank1_enable_set	cpu1_idata_bank0_enable_set	cpu1_itag_ram3_enable_set	cpu1_itag_ram2_enable_set	cpu1_itag_ram1_enable_set	cpu1_itag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-84. sec\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu1_ks_vim_ramecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ks_vim_ramecc_pend
26	b1tcm1_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm1_bank1_pend
25	b1tcm1_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm1_bank0_pend
24	b0tcm1_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm1_bank1_pend
23	b0tcm1_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm1_bank0_pend
22	atcm1_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm1_bank1_pend
21	atcm1_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm1_bank0_pend
20	cpu1_ddata_ram7_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram7_pend
19	cpu1_ddata_ram6_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram6_pend
18	cpu1_ddata_ram5_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram5_pend
17	cpu1_ddata_ram4_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram4_pend
16	cpu1_ddata_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram3_pend
15	cpu1_ddata_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram2_pend

**Table 20-84. sec\_enable\_set\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	cpu1_ddata_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram1_pend
13	cpu1_ddata_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram0_pend
12	cpu1_ddirty_ram_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddirty_ram_pend
11	cpu1_dtag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram3_pend
10	cpu1_dtag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram2_pend
9	cpu1_dtag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram1_pend
8	cpu1_dtag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram0_pend
7	cpu1_idata_bank3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank3_pend
6	cpu1_idata_bank2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank2_pend
5	cpu1_idata_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank1_pend
4	cpu1_idata_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank0_pend
3	cpu1_itag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram3_pend
2	cpu1_itag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram2_pend
1	cpu1_itag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram1_pend
0	cpu1_itag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram0_pend

#### 20.4.3.14 sec\_enable\_clr\_reg0 Register (Offset = C0h) [reset = X]

sec\_enable\_clr\_reg0 is shown in [Figure 20-56](#) and described in [Table 20-85](#).

Return to the [Table 20-71](#).

Interrupt Enable Clear Register 0

**Figure 20-56. sec\_enable\_clr\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu1_ks_vim_r amecc_enable_ clr	b1tcm1_bank1_ enable_clr	b1tcm1_bank0_ enable_clr	b0tcm1_bank1_ enable_clr
R/W-X				R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
b0tcm1_bank0_ enable_clr	atcm1_bank1_e nable_clr	atcm1_bank0_e nable_clr	cpu1_ddata_ra m7_enable_clr	cpu1_ddata_ra m6_enable_clr	cpu1_ddata_ra m5_enable_clr	cpu1_ddata_ra m4_enable_clr	cpu1_ddata_ra m3_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
cpu1_ddata_ra m2_enable_clr	cpu1_ddata_ra m1_enable_clr	cpu1_ddata_ra m0_enable_clr	cpu1_ddirty_ra m_enable_clr	cpu1_dtag_ram 3_enable_clr	cpu1_dtag_ram 2_enable_clr	cpu1_dtag_ram 1_enable_clr	cpu1_dtag_ram 0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
cpu1_idata_ban k3_enable_clr	cpu1_idata_ban k2_enable_clr	cpu1_idata_ban k1_enable_clr	cpu1_idata_ban k0_enable_clr	cpu1_itag_ram3 _enable_clr	cpu1_itag_ram2 _enable_clr	cpu1_itag_ram1 _enable_clr	cpu1_itag_ram0 _enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**Table 20-85. sec\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu1_ks_vim_ramecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ks_vim_ramecc_pend
26	b1tcm1_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm1_bank1_pend
25	b1tcm1_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm1_bank0_pend
24	b0tcm1_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm1_bank1_pend
23	b0tcm1_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm1_bank0_pend
22	atcm1_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm1_bank1_pend
21	atcm1_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm1_bank0_pend
20	cpu1_ddata_ram7_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram7_pend
19	cpu1_ddata_ram6_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram6_pend
18	cpu1_ddata_ram5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram5_pend
17	cpu1_ddata_ram4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram4_pend
16	cpu1_ddata_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram3_pend
15	cpu1_ddata_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram2_pend
14	cpu1_ddata_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram1_pend

**Table 20-85. sec\_enable\_clr\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
13	cpu1_ddata_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram0_pend
12	cpu1_ddirty_ram_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddirty_ram_pend
11	cpu1_dtag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram3_pend
10	cpu1_dtag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram2_pend
9	cpu1_dtag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram1_pend
8	cpu1_dtag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram0_pend
7	cpu1_idata_bank3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank3_pend
6	cpu1_idata_bank2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank2_pend
5	cpu1_idata_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank1_pend
4	cpu1_idata_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank0_pend
3	cpu1_itag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram3_pend
2	cpu1_itag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram2_pend
1	cpu1_itag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram1_pend
0	cpu1_itag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram0_pend



### 20.4.3.15 ded\_eoi\_reg Register (Offset = 13Ch) [reset = X]

ded\_eoi\_reg is shown in [Figure 20-57](#) and described in [Table 20-86](#).

Return to the [Table 20-71](#).

EOI Register

**Figure 20-57. ded\_eoi\_reg Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-X							R/W1S-0h

**Table 20-86. ded\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

### 20.4.3.16 ded\_status\_reg0 Register (Offset = 140h) [reset = X]

ded\_status\_reg0 is shown in [Figure 20-58](#) and described in [Table 20-87](#).

Return to the [Table 20-71](#).

Interrupt Status Register 0

**Figure 20-58. ded\_status\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu1_ks_vim_r amecc_pend	b1tcm1_bank1_ pend	b1tcm1_bank0_ pend	b0tcm1_bank1_ pend
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm1_bank0_ pend	atcm1_bank1_p end	atcm1_bank0_p end	cpu1_ddata_ra m7_pend	cpu1_ddata_ra m6_pend	cpu1_ddata_ra m5_pend	cpu1_ddata_ra m4_pend	cpu1_ddata_ra m3_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu1_ddata_ra m2_pend	cpu1_ddata_ra m1_pend	cpu1_ddata_ra m0_pend	cpu1_ddirty_ra m_pend	cpu1_dtag_ram 3_pend	cpu1_dtag_ram 2_pend	cpu1_dtag_ram 1_pend	cpu1_dtag_ram 0_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu1_idata_ban k3_pend	cpu1_idata_ban k2_pend	cpu1_idata_ban k1_pend	cpu1_idata_ban k0_pend	cpu1_itag_ram3 _pend	cpu1_itag_ram2 _pend	cpu1_itag_ram1 _pend	cpu1_itag_ram0 _pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-87. ded\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu1_ks_vim_ramecc_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ks_vim_ramecc_pend
26	b1tcm1_bank1_pend	R/W1S	0h	Interrupt Pending Status for b1tcm1_bank1_pend
25	b1tcm1_bank0_pend	R/W1S	0h	Interrupt Pending Status for b1tcm1_bank0_pend
24	b0tcm1_bank1_pend	R/W1S	0h	Interrupt Pending Status for b0tcm1_bank1_pend
23	b0tcm1_bank0_pend	R/W1S	0h	Interrupt Pending Status for b0tcm1_bank0_pend
22	atcm1_bank1_pend	R/W1S	0h	Interrupt Pending Status for atcm1_bank1_pend
21	atcm1_bank0_pend	R/W1S	0h	Interrupt Pending Status for atcm1_bank0_pend
20	cpu1_ddata_ram7_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram7_pend
19	cpu1_ddata_ram6_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram6_pend
18	cpu1_ddata_ram5_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram5_pend
17	cpu1_ddata_ram4_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram4_pend
16	cpu1_ddata_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram3_pend
15	cpu1_ddata_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram2_pend
14	cpu1_ddata_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram1_pend
13	cpu1_ddata_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddata_ram0_pend
12	cpu1_ddirty_ram_pend	R/W1S	0h	Interrupt Pending Status for cpu1_ddirty_ram_pend
11	cpu1_dtag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram3_pend
10	cpu1_dtag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram2_pend

**Table 20-87. ded\_status\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9	cpu1_dtag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram1_pend
8	cpu1_dtag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_dtag_ram0_pend
7	cpu1_idata_bank3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank3_pend
6	cpu1_idata_bank2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank2_pend
5	cpu1_idata_bank1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank1_pend
4	cpu1_idata_bank0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_idata_bank0_pend
3	cpu1_itag_ram3_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram3_pend
2	cpu1_itag_ram2_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram2_pend
1	cpu1_itag_ram1_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram1_pend
0	cpu1_itag_ram0_pend	R/W1S	0h	Interrupt Pending Status for cpu1_itag_ram0_pend

### 20.4.3.17 ded\_enable\_set\_reg0 Register (Offset = 180h) [reset = X]

ded\_enable\_set\_reg0 is shown in [Figure 20-59](#) and described in [Table 20-88](#).

Return to the [Table 20-71](#).

Interrupt Enable Set Register 0

**Figure 20-59. ded\_enable\_set\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu1_ks_vim_ramecc_enable_set	b1tcm1_bank1_enable_set	b1tcm1_bank0_enable_set	b0tcm1_bank1_enable_set
R/W-X				R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
23	22	21	20	19	18	17	16
b0tcm1_bank0_enable_set	atcm1_bank1_enable_set	atcm1_bank0_enable_set	cpu1_ddata_ram7_enable_set	cpu1_ddata_ram6_enable_set	cpu1_ddata_ram5_enable_set	cpu1_ddata_ram4_enable_set	cpu1_ddata_ram3_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
15	14	13	12	11	10	9	8
cpu1_ddata_ram2_enable_set	cpu1_ddata_ram1_enable_set	cpu1_ddata_ram0_enable_set	cpu1_ddirty_ram_enable_set	cpu1_dtag_ram3_enable_set	cpu1_dtag_ram2_enable_set	cpu1_dtag_ram1_enable_set	cpu1_dtag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h
7	6	5	4	3	2	1	0
cpu1_idata_bank3_enable_set	cpu1_idata_bank2_enable_set	cpu1_idata_bank1_enable_set	cpu1_idata_bank0_enable_set	cpu1_itag_ram3_enable_set	cpu1_itag_ram2_enable_set	cpu1_itag_ram1_enable_set	cpu1_itag_ram0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-88. ded\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu1_ks_vim_ramecc_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ks_vim_ramecc_pend
26	b1tcm1_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm1_bank1_pend
25	b1tcm1_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b1tcm1_bank0_pend
24	b0tcm1_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm1_bank1_pend
23	b0tcm1_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for b0tcm1_bank0_pend
22	atcm1_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm1_bank1_pend
21	atcm1_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for atcm1_bank0_pend
20	cpu1_ddata_ram7_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram7_pend
19	cpu1_ddata_ram6_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram6_pend
18	cpu1_ddata_ram5_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram5_pend
17	cpu1_ddata_ram4_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram4_pend
16	cpu1_ddata_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram3_pend
15	cpu1_ddata_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram2_pend

**Table 20-88. ded\_enable\_set\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
14	cpu1_ddata_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram1_pend
13	cpu1_ddata_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddata_ram0_pend
12	cpu1_ddirty_ram_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_ddirty_ram_pend
11	cpu1_dtag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram3_pend
10	cpu1_dtag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram2_pend
9	cpu1_dtag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram1_pend
8	cpu1_dtag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_dtag_ram0_pend
7	cpu1_idata_bank3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank3_pend
6	cpu1_idata_bank2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank2_pend
5	cpu1_idata_bank1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank1_pend
4	cpu1_idata_bank0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_idata_bank0_pend
3	cpu1_itag_ram3_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram3_pend
2	cpu1_itag_ram2_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram2_pend
1	cpu1_itag_ram1_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram1_pend
0	cpu1_itag_ram0_enable_set	R/W1S	0h	Interrupt Enable Set Register for cpu1_itag_ram0_pend

### 20.4.3.18 ded\_enable\_clr\_reg0 Register (Offset = 1C0h) [reset = X]

ded\_enable\_clr\_reg0 is shown in [Figure 20-60](#) and described in [Table 20-89](#).

Return to the [Table 20-71](#).

Interrupt Enable Clear Register 0

**Figure 20-60. ded\_enable\_clr\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED				cpu1_ks_vim_ramecc_enable_clr	b1tcm1_bank1_enable_clr	b1tcm1_bank0_enable_clr	b0tcm1_bank1_enable_clr
R/W-X				R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
23	22	21	20	19	18	17	16
b0tcm1_bank0_enable_clr	atcm1_bank1_enable_clr	atcm1_bank0_enable_clr	cpu1_ddata_ram7_enable_clr	cpu1_ddata_ram6_enable_clr	cpu1_ddata_ram5_enable_clr	cpu1_ddata_ram4_enable_clr	cpu1_ddata_ram3_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
15	14	13	12	11	10	9	8
cpu1_ddata_ram2_enable_clr	cpu1_ddata_ram1_enable_clr	cpu1_ddata_ram0_enable_clr	cpu1_ddirty_ram_enable_clr	cpu1_dtag_ram3_enable_clr	cpu1_dtag_ram2_enable_clr	cpu1_dtag_ram1_enable_clr	cpu1_dtag_ram0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h
7	6	5	4	3	2	1	0
cpu1_idata_bank3_enable_clr	cpu1_idata_bank2_enable_clr	cpu1_idata_bank1_enable_clr	cpu1_idata_bank0_enable_clr	cpu1_itag_ram3_enable_clr	cpu1_itag_ram2_enable_clr	cpu1_itag_ram1_enable_clr	cpu1_itag_ram0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**Table 20-89. ded\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	RESERVED	R/W	X	
27	cpu1_ks_vim_ramecc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ks_vim_ramecc_pend
26	b1tcm1_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm1_bank1_pend
25	b1tcm1_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b1tcm1_bank0_pend
24	b0tcm1_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm1_bank1_pend
23	b0tcm1_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for b0tcm1_bank0_pend
22	atcm1_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm1_bank1_pend
21	atcm1_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for atcm1_bank0_pend
20	cpu1_ddata_ram7_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram7_pend
19	cpu1_ddata_ram6_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram6_pend
18	cpu1_ddata_ram5_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram5_pend
17	cpu1_ddata_ram4_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram4_pend
16	cpu1_ddata_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram3_pend
15	cpu1_ddata_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram2_pend
14	cpu1_ddata_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram1_pend

**Table 20-89. ded\_enable\_clr\_reg0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
13	cpu1_ddata_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddata_ram0_pend
12	cpu1_ddirty_ram_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_ddirty_ram_pend
11	cpu1_dtag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram3_pend
10	cpu1_dtag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram2_pend
9	cpu1_dtag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram1_pend
8	cpu1_dtag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_dtag_ram0_pend
7	cpu1_idata_bank3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank3_pend
6	cpu1_idata_bank2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank2_pend
5	cpu1_idata_bank1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank1_pend
4	cpu1_idata_bank0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_idata_bank0_pend
3	cpu1_itag_ram3_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram3_pend
2	cpu1_itag_ram2_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram2_pend
1	cpu1_itag_ram1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram1_pend
0	cpu1_itag_ram0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for cpu1_itag_ram0_pend

### 20.4.3.19 aggr\_enable\_set Register (Offset = 200h) [reset = X]

aggr\_enable\_set is shown in [Figure 20-61](#) and described in [Table 20-90](#).

Return to the [Table 20-71](#).

AGGR interrupt enable set Register

**Figure 20-61. aggr\_enable\_set Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-X						R/W1S-0h	R/W1S-0h

**Table 20-90. aggr\_enable\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1S	0h	interrupt enable set for svbus timeout errors
0	parity	R/W1S	0h	interrupt enable set for parity errors



### 20.4.3.20 aggr\_enable\_clr Register (Offset = 204h) [reset = X]

aggr\_enable\_clr is shown in [Figure 20-62](#) and described in [Table 20-91](#).

Return to the [Table 20-71](#).

AGGR interrupt enable clear Register

**Figure 20-62. aggr\_enable\_clr Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-X						R/W1C-0h	R/W1C-0h

**Table 20-91. aggr\_enable\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1C	0h	interrupt enable clear for svbus timeout errors
0	parity	R/W1C	0h	interrupt enable clear for parity errors

### 20.4.3.21 aggr\_status\_set Register (Offset = 208h) [reset = X]

aggr\_status\_set is shown in [Figure 20-63](#) and described in [Table 20-92](#).

Return to the [Table 20-71](#).

AGGR interrupt status set Register

**Figure 20-63. aggr\_status\_set Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				timeout		parity	
R/W-X				R/Wincr-0h		R/Wincr-0h	

**Table 20-92. aggr\_status\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wincr	0h	interrupt status set for svbus timeout errors
1-0	parity	R/Wincr	0h	interrupt status set for parity errors

### 20.4.3.22 aggr\_status\_clr Register (Offset = 20Ch) [reset = X]

aggr\_status\_clr is shown in [Figure 20-64](#) and described in [Table 20-93](#).

Return to the [Table 20-71](#).

AGGR interrupt status clear Register

**Figure 20-64. aggr\_status\_clr Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				timeout		parity	
R/W-X				R/Wdecr-0h		R/Wdecr-0h	

**Table 20-93. aggr\_status\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wdecr	0h	interrupt status clear for svbus timeout errors
1-0	parity	R/Wdecr	0h	interrupt status clear for parity errors

## 20.4.4 MSS\_ECC\_AGG\_MSS Registers

Table 20-94 lists the MSS\_ECC\_AGG\_MSS registers. All register offset addresses not listed in Table 20-94 should be considered as reserved locations and the register contents should not be modified.

**Table 20-94. MSS\_ECC\_AGG\_MSS Registers**

Offset	Acronym	Register Name	Section
0h	rev	Aggregator Revision Register	<a href="#">Section 20.4.4.1</a>
8h	vector	ECC Vector Register	<a href="#">Section 20.4.4.2</a>
Ch	stat	Misc Status	<a href="#">Section 20.4.4.3</a>
10h	wrap_rev	ECC Wrapper Revision Register	<a href="#">Section 20.4.4.4</a>
14h	ctrl	ECC Control	<a href="#">Section 20.4.4.5</a>
18h	err_ctrl1	ECC Error Control1 Register	<a href="#">Section 20.4.4.6</a>
1Ch	err_ctrl2	ECC Error Control2 Register	<a href="#">Section 20.4.4.7</a>
20h	err_stat1	ECC Error Status1 Register	<a href="#">Section 20.4.4.8</a>
24h	err_stat2	ECC Error Status2 Register	<a href="#">Section 20.4.4.9</a>
28h	err_stat3	ECC Error Status3 Register	<a href="#">Section 20.4.4.10</a>
3Ch	sec_eoi_reg	EOI Register	<a href="#">Section 20.4.4.11</a>
40h	sec_status_reg0	Interrupt Status Register 0	<a href="#">Section 20.4.4.12</a>
80h	sec_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Section 20.4.4.13</a>
C0h	sec_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Section 20.4.4.14</a>
13Ch	ded_eoi_reg	EOI Register	<a href="#">Section 20.4.4.15</a>
140h	ded_status_reg0	Interrupt Status Register 0	<a href="#">Section 20.4.4.16</a>
180h	ded_enable_set_reg0	Interrupt Enable Set Register 0	<a href="#">Section 20.4.4.17</a>
1C0h	ded_enable_clr_reg0	Interrupt Enable Clear Register 0	<a href="#">Section 20.4.4.18</a>
200h	aggr_enable_set	AGGR interrupt enable set Register	<a href="#">Section 20.4.4.19</a>
204h	aggr_enable_clr	AGGR interrupt enable clear Register	<a href="#">Section 20.4.4.20</a>
208h	aggr_status_set	AGGR interrupt status set Register	<a href="#">Section 20.4.4.21</a>
20Ch	aggr_status_clr	AGGR interrupt status clear Register	<a href="#">Section 20.4.4.22</a>

#### 20.4.4.1 rev Register (Offset = 0h) [reset = 66A0C200h]

rev is shown in [Figure 20-65](#) and described in [Table 20-95](#).

Return to the [Table 20-94](#).

Revision parameters

**Figure 20-65. rev Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu			module_id										
R-1h			R-2h			R-6A0h									
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-18h						R-2h		R-0h			R-0h				

**Table 20-95. rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A0h	Module ID
15-11	revrtl	R	18h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	0h	Minor version

### 20.4.4.2 vector Register (Offset = 8h) [reset = X]

vector is shown in [Figure 20-66](#) and described in [Table 20-96](#).

Return to the [Table 20-94](#).

ECC Vector Register

**Figure 20-66. vector Register**

31	30	29	28	27	26	25	24
RESERVED							rd_svbus_done
R/W-X							R-0h
23	22	21	20	19	18	17	16
rd_svbus_address							
R/W-0h							
15	14	13	12	11	10	9	8
rd_svbus	RESERVED					ecc_vector	
R/W1S-0h	R/W-X					R/W-0h	
7	6	5	4	3	2	1	0
ecc_vector							
R/W-0h							

**Table 20-96. vector Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R/W	X	
24	rd_svbus_done	R	0h	Status to indicate if read on serial VBUS is complete
23-16	rd_svbus_address	R/W	0h	Read address
15	rd_svbus	R/W1S	0h	Write 1 to trigger a read on the serial VBUS
14-11	RESERVED	R/W	X	
10-0	ecc_vector	R/W	0h	Value written to select the corresponding ECC RAM for control or status

### 20.4.4.3 stat Register (Offset = Ch) [reset = X]

stat is shown in [Figure 20-67](#) and described in [Table 20-97](#).

Return to the [Table 20-94](#).

Misc Status

**Figure 20-67. stat Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											num_rams																				
R-X											R-8h																				

**Table 20-97. stat Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-11	RESERVED	R	X	
10-0	num_rams	R	8h	Indicates the number of RAMS serviced by the ECC aggregator

#### 20.4.4.4 wrap\_rev Register (Offset = 10h) [reset = 66A40202h]

wrap\_rev is shown in [Figure 20-68](#) and described in [Table 20-98](#).

Return to the [Table 20-94](#).

Revision parameters

**Figure 20-68. wrap\_rev Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
scheme		bu		module_id											
R-1h		R-2h		R-6A4h											
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
revrtl					revmaj			custom		revmin					
R-0h					R-2h			R-0h		R-2h					

**Table 20-98. wrap\_rev Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	scheme	R	1h	Scheme
29-28	bu	R	2h	bu
27-16	module_id	R	6A4h	Module ID
15-11	revrtl	R	0h	RTL version
10-8	revmaj	R	2h	Major version
7-6	custom	R	0h	Custom version
5-0	revmin	R	2h	Minor version



#### 20.4.4.5 ctrl Register (Offset = 14h) [reset = X]

ctrl is shown in [Figure 20-69](#) and described in [Table 20-99](#).

Return to the [Table 20-94](#).

ECC Control Register

**Figure 20-69. ctrl Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							check_svbus_timeout
R/W-X							R/W-1h
7	6	5	4	3	2	1	0
check_parity	error_once	force_n_row	force_ded	force_sec	enable_rmw	ecc_check	ecc_enable
R/W-1h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-1h	R/W-1h	R/W-1h

**Table 20-99. ctrl Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	RESERVED	R/W	X	
8	check_svbus_timeout	R/W	1h	check for svbus timeout errors
7	check_parity	R/W	1h	check for parity errors
6	error_once	R/W	0h	Force Error only once
5	force_n_row	R/W	0h	Force Error on any RAM read
4	force_ded	R/W	0h	Force Double Bit Error
3	force_sec	R/W	0h	Force Single Bit Error
2	enable_rmw	R/W	1h	Enable rmw
1	ecc_check	R/W	1h	Enable ECC check
0	ecc_enable	R/W	1h	Enable ECC

#### 20.4.4.6 err\_ctrl1 Register (Offset = 18h) [reset = 0h]

err\_ctrl1 is shown in [Figure 20-70](#) and described in [Table 20-100](#).

Return to the [Table 20-94](#).

ECC Error Control1 Register

**Figure 20-70. err\_ctrl1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_row																															
R/W-0h																															

**Table 20-100. err\_ctrl1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R/W	0h	Row address where single or double-bit error needs to be applied. This is ignored if force_n_row is set

#### 20.4.4.7 err\_ctrl2 Register (Offset = 1Ch) [reset = 0h]

err\_ctrl2 is shown in [Figure 20-71](#) and described in [Table 20-101](#).

Return to the [Table 20-94](#).

ECC Error Control2 Register

**Figure 20-71. err\_ctrl2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_bit2																ecc_bit1															
R/W-0h																R/W-0h															

**Table 20-101. err\_ctrl2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit2	R/W	0h	Data bit that needs to be flipped if double bit error needs to be forced
15-0	ecc_bit1	R/W	0h	Data bit that needs to be flipped when force_sec is set

#### 20.4.4.8 err\_stat1 Register (Offset = 20h) [reset = 0h]

err\_stat1 is shown in [Figure 20-72](#) and described in [Table 20-102](#).

Return to the [Table 20-94](#).

ECC Error Status1 Register

**Figure 20-72. err\_stat1 Register**

31	30	29	28	27	26	25	24
ecc_bit1							
R-0h							
23	22	21	20	19	18	17	16
ecc_bit1							
R-0h							
15	14	13	12	11	10	9	8
clr_ctrl_reg_err	clr_parity_err		clr_ecc_other	clr_ecc_ded		clr_ecc_sec	
R/W1C-0h	R/Wdecr-0h		R/W1C-0h	R/Wdecr-0h		R/Wdecr-0h	
7	6	5	4	3	2	1	0
ctr_reg_err	parity_err		ecc_other	ecc_ded		ecc_sec	
R/W1S-0h	R/W1S-0h		R/W1S-0h	R/Wincr-0h		R/Wincr-0h	

**Table 20-102. err\_stat1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	ecc_bit1	R	0h	Data bit that corresponds to the single-bit error
15	clr_ctrl_reg_err	R/W1C	0h	Clear control reg error Error Status, you must also re write the control register itself to clear this
14-13	clr_parity_err	R/Wdecr	0h	Clear parity Error Status
12	clr_ecc_other	R/W1C	0h	Clear other Error Status
11-10	clr_ecc_ded	R/Wdecr	0h	Clear Double Bit Error Status
9-8	clr_ecc_sec	R/Wdecr	0h	Clear Single Bit Error Status
7	ctr_reg_err	R/W1S	0h	control register error pending, Level interrupt
6-5	parity_err	R/W1S	0h	Level parity error Error Status
4	ecc_other	R/W1S	0h	successive single-bit errors have occurred while a writeback is still pending, Level interrupt
3-2	ecc_ded	R/Wincr	0h	Level Double Bit Error Status
1-0	ecc_sec	R/Wincr	0h	Level Single Bit Error Status

#### 20.4.4.9 err\_stat2 Register (Offset = 24h) [reset = 0h]

err\_stat2 is shown in [Figure 20-73](#) and described in [Table 20-103](#).

Return to the [Table 20-94](#).

ECC Error Status2 Register

**Figure 20-73. err\_stat2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ecc_row																															
R-0h																															

**Table 20-103. err\_stat2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ecc_row	R	0h	Row address where the single or double-bit error has occurred

#### 20.4.4.10 err\_stat3 Register (Offset = 28h) [reset = X]

err\_stat3 is shown in [Figure 20-74](#) and described in [Table 20-104](#).

Return to the [Table 20-94](#).

ECC Error Status3 Register

**Figure 20-74. err\_stat3 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED						clr_svbus_timeo ut_err	RESERVED
R/W-X						R/W1C-0h	R/W-X
7	6	5	4	3	2	1	0
RESERVED						svbus_timeout_ err	wb_pend
R/W-X						R/W1S-0h	R-0h

**Table 20-104. err\_stat3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	RESERVED	R/W	X	
9	clr_svbus_timeout_err	R/W1C	0h	Clear svbus timeout Error Status
8-2	RESERVED	R/W	X	
1	svbus_timeout_err	R/W1S	0h	Level svbus timeout error Error Status
0	wb_pend	R	0h	delayed write back pending Status

#### 20.4.4.11 sec\_eoi\_reg Register (Offset = 3Ch) [reset = X]

sec\_eoi\_reg is shown in [Figure 20-75](#) and described in [Table 20-105](#).

Return to the [Table 20-94](#).

EOI Register

**Figure 20-75. sec\_eoi\_reg Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-X							R/W1S-0h

**Table 20-105. sec\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

#### 20.4.4.12 sec\_status\_reg0 Register (Offset = 40h) [reset = X]

sec\_status\_reg0 is shown in [Figure 20-76](#) and described in [Table 20-106](#).

Return to the [Table 20-94](#).

Interrupt Status Register 0

**Figure 20-76. sec\_status\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
tptc_b0_pend	tptc_a1_pend	tptc_a0_pend	gpadc_pend	mss_retram_pend	mss_mbox_pend	mss_l2slv1_pend	mss_l2slv0_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-106. sec\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7	tptc_b0_pend	R/W1S	0h	Interrupt Pending Status for tptc_b0_pend
6	tptc_a1_pend	R/W1S	0h	Interrupt Pending Status for tptc_a1_pend
5	tptc_a0_pend	R/W1S	0h	Interrupt Pending Status for tptc_a0_pend
4	gpadc_pend	R/W1S	0h	Interrupt Pending Status for gpadc_pend
3	mss_retram_pend	R/W1S	0h	Interrupt Pending Status for mss_retram_pend
2	mss_mbox_pend	R/W1S	0h	Interrupt Pending Status for mss_mbox_pend
1	mss_l2slv1_pend	R/W1S	0h	Interrupt Pending Status for mss_l2slv1_pend
0	mss_l2slv0_pend	R/W1S	0h	Interrupt Pending Status for mss_l2slv0_pend



### 20.4.4.13 sec\_enable\_set\_reg0 Register (Offset = 80h) [reset = X]

sec\_enable\_set\_reg0 is shown in [Figure 20-77](#) and described in [Table 20-107](#).

Return to the [Table 20-94](#).

Interrupt Enable Set Register 0

**Figure 20-77. sec\_enable\_set\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
tptc_b0_enable_set	tptc_a1_enable_set	tptc_a0_enable_set	gpadc_enable_set	mss_retram_enable_set	mss_mbox_enable_set	mss_l2slv1_enable_set	mss_l2slv0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-107. sec\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7	tptc_b0_enable_set	R/W1S	0h	Interrupt Enable Set Register for tptc_b0_pend
6	tptc_a1_enable_set	R/W1S	0h	Interrupt Enable Set Register for tptc_a1_pend
5	tptc_a0_enable_set	R/W1S	0h	Interrupt Enable Set Register for tptc_a0_pend
4	gpadc_enable_set	R/W1S	0h	Interrupt Enable Set Register for gpadc_pend
3	mss_retram_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_retram_pend
2	mss_mbox_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_mbox_pend
1	mss_l2slv1_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_l2slv1_pend
0	mss_l2slv0_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_l2slv0_pend

#### 20.4.4.14 sec\_enable\_clr\_reg0 Register (Offset = C0h) [reset = X]

sec\_enable\_clr\_reg0 is shown in [Figure 20-78](#) and described in [Table 20-108](#).

Return to the [Table 20-94](#).

Interrupt Enable Clear Register 0

**Figure 20-78. sec\_enable\_clr\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
tptc_b0_enable_clr	tptc_a1_enable_clr	tptc_a0_enable_clr	gpadc_enable_clr	mss_retram_enable_clr	mss_mbox_enable_clr	mss_l2slv1_enable_clr	mss_l2slv0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**Table 20-108. sec\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7	tptc_b0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for tptc_b0_pend
6	tptc_a1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for tptc_a1_pend
5	tptc_a0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for tptc_a0_pend
4	gpadc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for gpadc_pend
3	mss_retram_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_retram_pend
2	mss_mbox_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_mbox_pend
1	mss_l2slv1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_l2slv1_pend
0	mss_l2slv0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_l2slv0_pend

#### 20.4.4.15 ded\_eoi\_reg Register (Offset = 13Ch) [reset = X]

ded\_eoi\_reg is shown in [Figure 20-79](#) and described in [Table 20-109](#).

Return to the [Table 20-94](#).

EOI Register

**Figure 20-79. ded\_eoi\_reg Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED							eoi_wr
R/W-X							R/W1S-0h

**Table 20-109. ded\_eoi\_reg Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	X	
0	eoi_wr	R/W1S	0h	EOI Register

#### 20.4.4.16 ded\_status\_reg0 Register (Offset = 140h) [reset = X]

ded\_status\_reg0 is shown in [Figure 20-80](#) and described in [Table 20-110](#).

Return to the [Table 20-94](#).

Interrupt Status Register 0

**Figure 20-80. ded\_status\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
tptc_b0_pend	tptc_a1_pend	tptc_a0_pend	gpadc_pend	mss_retram_pend	mss_mbox_pend	mss_l2slv1_pend	mss_l2slv0_pend
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-110. ded\_status\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7	tptc_b0_pend	R/W1S	0h	Interrupt Pending Status for tptc_b0_pend
6	tptc_a1_pend	R/W1S	0h	Interrupt Pending Status for tptc_a1_pend
5	tptc_a0_pend	R/W1S	0h	Interrupt Pending Status for tptc_a0_pend
4	gpadc_pend	R/W1S	0h	Interrupt Pending Status for gpadc_pend
3	mss_retram_pend	R/W1S	0h	Interrupt Pending Status for mss_retram_pend
2	mss_mbox_pend	R/W1S	0h	Interrupt Pending Status for mss_mbox_pend
1	mss_l2slv1_pend	R/W1S	0h	Interrupt Pending Status for mss_l2slv1_pend
0	mss_l2slv0_pend	R/W1S	0h	Interrupt Pending Status for mss_l2slv0_pend

#### 20.4.4.17 ded\_enable\_set\_reg0 Register (Offset = 180h) [reset = X]

ded\_enable\_set\_reg0 is shown in [Figure 20-81](#) and described in [Table 20-111](#).

Return to the [Table 20-94](#).

Interrupt Enable Set Register 0

**Figure 20-81. ded\_enable\_set\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
tptc_b0_enable_set	tptc_a1_enable_set	tptc_a0_enable_set	gpadc_enable_set	mss_retram_enable_set	mss_mbox_enable_set	mss_l2slv1_enable_set	mss_l2slv0_enable_set
R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h	R/W1S-0h

**Table 20-111. ded\_enable\_set\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7	tptc_b0_enable_set	R/W1S	0h	Interrupt Enable Set Register for tptc_b0_pend
6	tptc_a1_enable_set	R/W1S	0h	Interrupt Enable Set Register for tptc_a1_pend
5	tptc_a0_enable_set	R/W1S	0h	Interrupt Enable Set Register for tptc_a0_pend
4	gpadc_enable_set	R/W1S	0h	Interrupt Enable Set Register for gpadc_pend
3	mss_retram_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_retram_pend
2	mss_mbox_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_mbox_pend
1	mss_l2slv1_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_l2slv1_pend
0	mss_l2slv0_enable_set	R/W1S	0h	Interrupt Enable Set Register for mss_l2slv0_pend

#### 20.4.4.18 ded\_enable\_clr\_reg0 Register (Offset = 1C0h) [reset = X]

ded\_enable\_clr\_reg0 is shown in [Figure 20-82](#) and described in [Table 20-112](#).

Return to the [Table 20-94](#).

Interrupt Enable Clear Register 0

**Figure 20-82. ded\_enable\_clr\_reg0 Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
tptc_b0_enable_clr	tptc_a1_enable_clr	tptc_a0_enable_clr	gpadc_enable_clr	mss_retram_enable_clr	mss_mbox_enable_clr	mss_l2slv1_enable_clr	mss_l2slv0_enable_clr
R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h	R/W1C-0h

**Table 20-112. ded\_enable\_clr\_reg0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R/W	X	
7	tptc_b0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for tptc_b0_pend
6	tptc_a1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for tptc_a1_pend
5	tptc_a0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for tptc_a0_pend
4	gpadc_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for gpadc_pend
3	mss_retram_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_retram_pend
2	mss_mbox_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_mbox_pend
1	mss_l2slv1_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_l2slv1_pend
0	mss_l2slv0_enable_clr	R/W1C	0h	Interrupt Enable Clear Register for mss_l2slv0_pend

#### 20.4.4.19 aggr\_enable\_set Register (Offset = 200h) [reset = X]

aggr\_enable\_set is shown in [Figure 20-83](#) and described in [Table 20-113](#).

Return to the [Table 20-94](#).

AGGR interrupt enable set Register

**Figure 20-83. aggr\_enable\_set Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-X						R/W1S-0h	R/W1S-0h

**Table 20-113. aggr\_enable\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1S	0h	interrupt enable set for svbus timeout errors
0	parity	R/W1S	0h	interrupt enable set for parity errors

#### 20.4.4.20 aggr\_enable\_clr Register (Offset = 204h) [reset = X]

aggr\_enable\_clr is shown in [Figure 20-84](#) and described in [Table 20-114](#).

Return to the [Table 20-94](#).

AGGR interrupt enable clear Register

**Figure 20-84. aggr\_enable\_clr Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED						timeout	parity
R/W-X						R/W1C-0h	R/W1C-0h

**Table 20-114. aggr\_enable\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	RESERVED	R/W	X	
1	timeout	R/W1C	0h	interrupt enable clear for svbus timeout errors
0	parity	R/W1C	0h	interrupt enable clear for parity errors



#### 20.4.4.21 aggr\_status\_set Register (Offset = 208h) [reset = X]

aggr\_status\_set is shown in [Figure 20-85](#) and described in [Table 20-115](#).

Return to the [Table 20-94](#).

AGGR interrupt status set Register

**Figure 20-85. aggr\_status\_set Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				timeout		parity	
R/W-X				R/Wincr-0h		R/Wincr-0h	

**Table 20-115. aggr\_status\_set Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wincr	0h	interrupt status set for svbus timeout errors
1-0	parity	R/Wincr	0h	interrupt status set for parity errors

#### 20.4.4.22 aggr\_status\_clr Register (Offset = 20Ch) [reset = X]

aggr\_status\_clr is shown in [Figure 20-86](#) and described in [Table 20-116](#).

Return to the [Table 20-94](#).

AGGR interrupt status clear Register

**Figure 20-86. aggr\_status\_clr Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-X							
23	22	21	20	19	18	17	16
RESERVED							
R/W-X							
15	14	13	12	11	10	9	8
RESERVED							
R/W-X							
7	6	5	4	3	2	1	0
RESERVED				timeout		parity	
R/W-X				R/Wdecr-0h		R/Wdecr-0h	

**Table 20-116. aggr\_status\_clr Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	X	
3-2	timeout	R/Wdecr	0h	interrupt status clear for svbus timeout errors
1-0	parity	R/Wdecr	0h	interrupt status clear for parity errors

## 20.5 Error Signaling Module (ESM)

This section provides the details of the error signaling module (ESM) that aggregates device errors and provides internal and external error response based on error severity.

### 20.5.1 Overview

The Error Signaling Module (ESM) collects and reports the various error conditions on the microcontroller. The error condition is categorized based on a severity level. Error response is then generated based on the category of the error. Possible error responses include a low priority interrupt, high priority interrupt, and an external pin action.

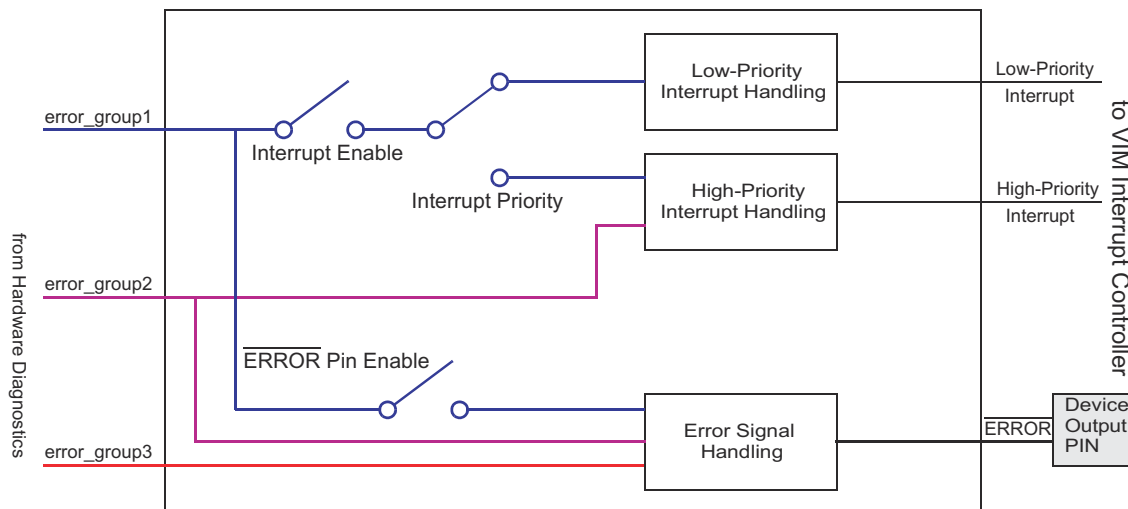
### 20.5.2 Feature List

- Up to 192 error channels (MSS/DSS) are supported, divided into 3 different groups:
  - 128 Group1 (low severity) channels with configurable interrupt generation and configurable  $\overline{\text{ERROR}}$  pin behavior
  - 32 Group2 (high severity) channels with predefined interrupt generation and predefined  $\overline{\text{ERROR}}$  pin behavior
  - 32 Group3 (high severity) channels with no interrupt generation and predefined  $\overline{\text{ERROR}}$  pin behavior. These channels have no interrupt response as they are reserved for CPU based diagnostics that generate aborts directly to the CPU.
- Dedicated device  $\overline{\text{ERROR}}$  pin to signal an external observer
- Configurable timebase for  $\overline{\text{ERROR}}$  pin output
- Error forcing capability for latent fault testing

### 20.5.3 Block Diagram

As shown in [Figure 20-87](#), the ESM channels are divided into three groups. Group1 channels are considered to be low severity. Group1 errors have a configurable interrupt response and configurable  $\overline{\text{ERROR}}$  pin behavior. Note that the ESM Status Register 1 (ESMSR1) for error group 1 gets updated, regardless if the interrupt enable is active or not. Group2 channels are  $\overline{\text{ERROR}}$  high severity. Group2 errors always generate a high priority interrupt and an output on the  $\overline{\text{ERROR}}$  pin. Group3 errors are reserved for high severity errors generated by diagnostics which have already generated a CPU abort response. Because an abort response is generated, there is no need to generate an interrupt response. Group3 errors always generate an  $\overline{\text{ERROR}}$  pin output.

The ESM interrupt and  $\overline{\text{ERROR}}$  pin behavior are also summarized in [Table 20-117](#).



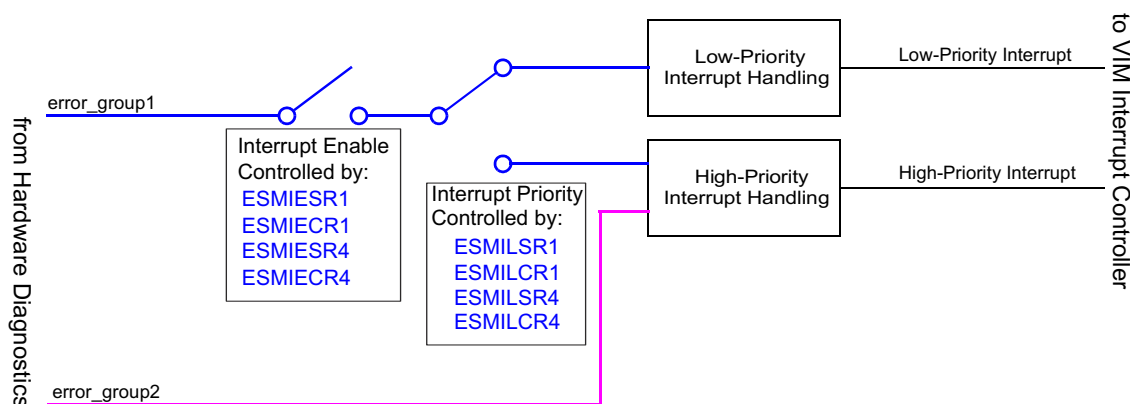
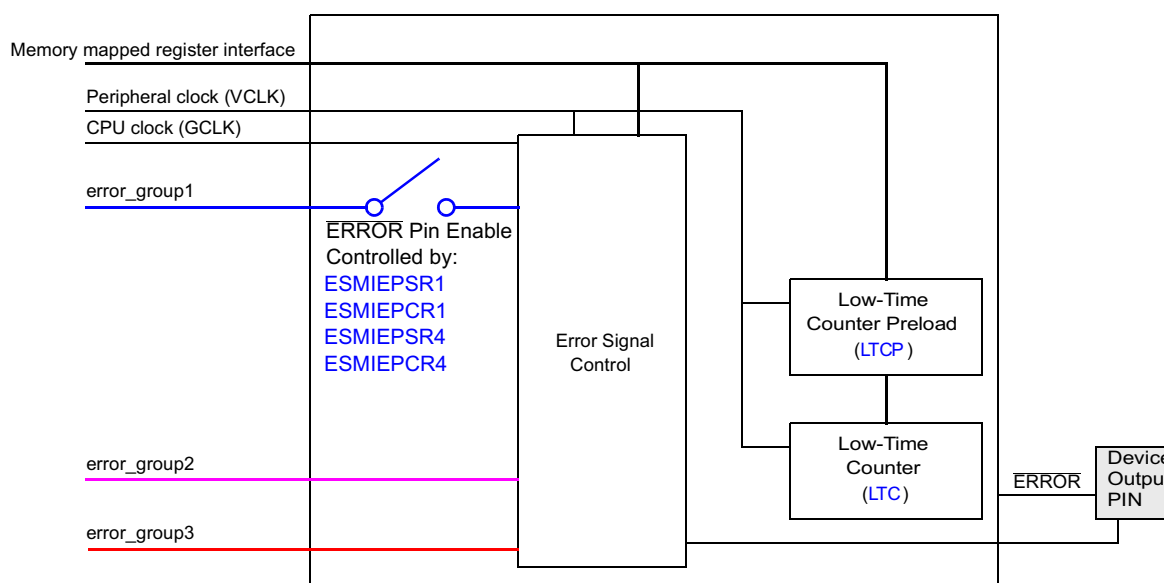
Note that the ESM Status Register 1 (ESMSR1) for error\_group1 gets updated, regardless if the interrupt enable is active or not.

**Figure 20-87. Block Diagram**

**Table 20-117. ESM Interrupt and  $\overline{\text{ERROR}}$  Pin Behavior**

Error Group	Interrupt Generated	Interrupt Priority	$\overline{\text{ERROR}}$ Pin Response Generated
1	configurable interrupt	configurable priority	configurable output generation
2	interrupt generated	high priority	output generated
3	no interrupt	NA	output generated

Figure 20-88 and Figure 20-89 show the interrupt response handling and  $\overline{\text{ERROR}}$  pin response handling with register configuration. The total active time of the  $\overline{\text{ERROR}}$  pin is controlled by the Low-Time Counter Preload register (LTCP) and the key register (ESMEPSR) as shown in Figure 20-89. See for details.


**Figure 20-88. Interrupt Response Handling**

**Figure 20-89.  $\overline{\text{ERROR}}$  Pin Response Handling**

## 20.5.4 Integration Details

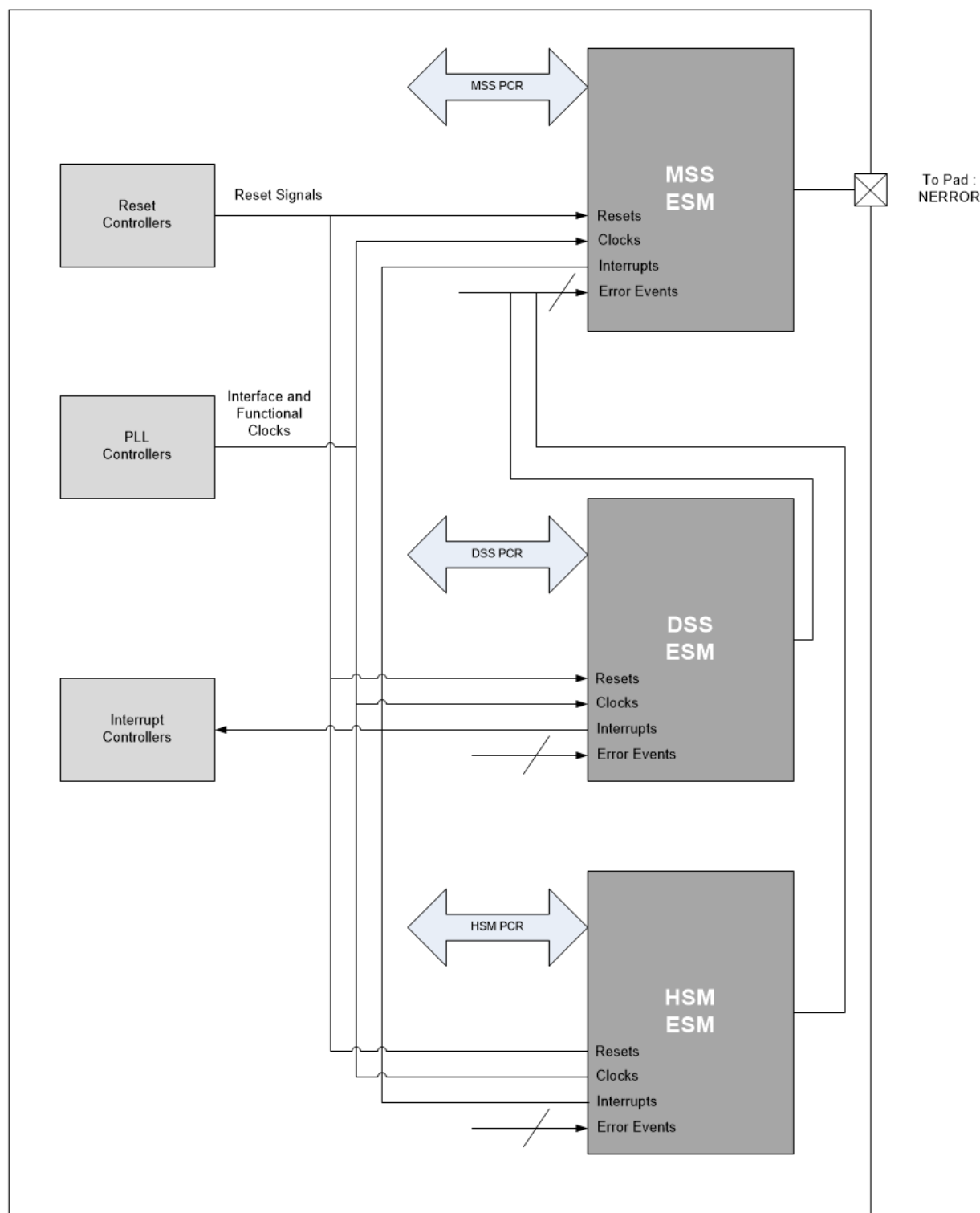
**Table 20-118. ESM Channel Distribution**

	Parameters
--	------------

**Table 20-118. ESM Channel Distribution (continued)**

Instance	Max Group 1	Max Group 2	Max Group 3
MSS_ESM	128	32	32
HSM_ESM	Refer HSM Design Specification		

Refer to [Section 20.3.9](#) for the ESM Interrupt map.



**Figure 20-90. ESM Block Diagram**

### 20.5.5 Module Operation

This device has 192 error channels (MSS & DSS), divided into 3 different error groups. Please refer to the device datasheet for ESM channel assignment details.

The ESM module has error flags for each error channel. The error status registers ESMSR1, ESMSR4, ESMSR2, ESMSR3 provide status information on a pending error of Group1 (Channel 0-31), Group1 (Channel 32-63), Group1 (Channel 64-95), Group2, and Group3, respectively. The ESMEPSR register provides the current ERROR status. The module also provides a status shadow register, ESMSSR2, which maintains the error flags of Group2 until power-on reset (PORRST) is asserted. See for details of their behavior during power on reset and warm reset.

Once an error occurs, the ESM module will set the corresponding error flags. In addition, it can trigger an interrupt, ERROR pin outputs low depending on the ESM settings. Once the ERROR pin outputs low, a power on reset or a write of 0x5 to ESMEKR is required to release the ESM error pin back to normal state. The application can read the error status registers (ESMSR1, ESMSR4, ESMSR7, ESMSR2, and ESMSR3) to debug the error. If an RST is triggered or the error interrupt has been served, the error flag of Group2 should be read from ESMSSR2 because the error flag in ESMSR2 will be cleared by RST. You can also test the functionality of the ERROR pin by forcing an error.

You can also test the functionality of the ERROR pin by forcing an error.

#### 20.5.5.1 Reset Behavior

Power on reset:

- **ERROR pin behavior**

When nPORRST is active, the  $\overline{\text{ERROR}}$  pin is in a high impedance state (output drivers disabled).

- **Register behavior**

After PORRST, all registers in ESM module will be re-initialized to the default value. All the error status registers are cleared to zero.

Warm reset ( $\overline{\text{RST}}$ ):

- **ERROR pin behavior**

During  $\overline{\text{RST}}$ , the  $\overline{\text{ERROR}}$  pin is in "output active" state with pull-down disabled. The  $\overline{\text{ERROR}}$  pin remains unchanged after  $\overline{\text{RST}}$ .

- **Register behavior**

After  $\overline{\text{RST}}$ , ESMSR1, ESMSR4, ESMSR7, ESMSSR2, ESMSR3 and ESMEPSR register values remains un-changed. Since  $\overline{\text{RST}}$  does not clear the critical failure registers, the user can read those registers to debug the failures after  $\overline{\text{RST}}$  pin goes back to high.

After  $\overline{\text{RST}}$ , if one of the flags in ESMSR1, ESMSR4 and ESMSR7 is set, the interrupt service routine will be called once the corresponding interrupt is enabled.

•

---

#### Note

ESMSR2 is cleared after  $\overline{\text{RST}}$ . The flag in ESMSR2 gets cleared when reading the appropriate vector in the ESMIOFFHR offset register. Reading ESMIOFFHR will not clear the ESMSR1, ESMSR4, ESMSR7 and the shadow register ESMSSR2. Reading ESMIOFFLR will also not clear the ESMSR1, ESMSR4 and ESMSR7.

---

### 20.5.5.2 $\overline{\text{ERROR}}$ Pin Timing

The  $\overline{\text{ERROR}}$  pin is an active low function. The state of the pin is also readable from  $\overline{\text{ERROR}}$  Pin Status Register (ESMEPSR). A warm reset ( $\overline{\text{RST}}$ ) does not affect the state of the pin. The pin is in a high-impedance state during power-on reset. Once the ESM module drives the  $\overline{\text{ERROR}}$  pin low, it remains in this state for the time specified by the Low-Time Counter Preload register (LTCPR). Based on the time period of the peripheral clock (VCLK), the total active time of the  $\overline{\text{ERROR}}$  pin can be calculated as:

$$t_{\overline{\text{ERROR\_low}}} = t_{\text{VCLK}} \times (\text{LTCP} + 1) \quad (8)$$

Once this period expires, the  $\overline{\text{ERROR}}$  pin is set to high in case the reset of the  $\overline{\text{ERROR}}$  pin was requested. This request is done by writing an appropriate key (0x5) to the key register (ESMEKR) during the  $\overline{\text{ERROR}}$  pin low time. Here are a few examples:

Example 1: ESM detects a failure and drives the  $\overline{\text{ERROR}}$  pin low. No  $\overline{\text{ERROR}}$  pin reset is requested. The  $\overline{\text{ERROR}}$  pin continues outputting low until power on reset occurs.



**Figure 20-91.  $\overline{\text{ERROR}}$  Pin Timing - Example 1**

Example 2: ESM detects a failure and drives the  $\overline{\text{ERROR}}$  pin low. An  $\overline{\text{ERROR}}$  pin reset request is received before  $t_{\overline{\text{ERROR\_low}}}$  expires. In this case, the  $\overline{\text{ERROR}}$  pin is set to high immediately after  $t_{\overline{\text{ERROR\_low}}}$  expires.



**Figure 20-92.  $\overline{\text{ERROR}}$  Pin Timing - Example 2**

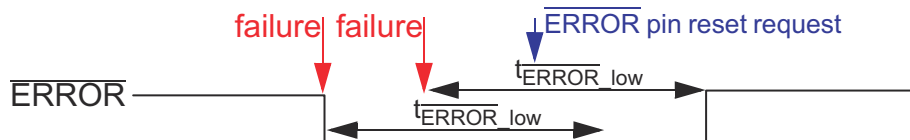
Example 3: ESM detects a failure and drives the  $\overline{\text{ERROR}}$  pin low. An  $\overline{\text{ERROR}}$  pin reset request is received after  $t_{\overline{\text{ERROR\_low}}}$  expires. In this case, the  $\overline{\text{ERROR}}$  pin is set to high immediately after  $\overline{\text{ERROR}}$  pin reset request is received.



**Figure 20-93.  $\overline{\text{ERROR}}$  Pin Timing - Example 3**

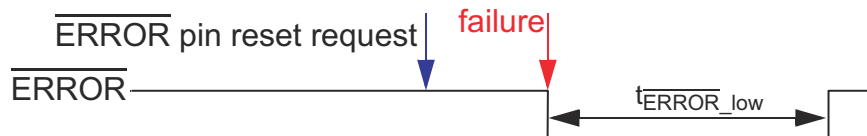


Example 4: ESM detects a failure and drives the  $\overline{\text{ERROR}}$  pin low. Another failure occurs within the time the pin stays low. In this case, the low time counter will be reset when the other failure occurs. In other words,  $t_{\text{ERROR\_low}}$  should be counted from whenever the most recent failure occurs.



**Figure 20-94.  $\overline{\text{ERROR}}$  Pin Timing - Example 4**

Example 5: The reset of the  $\overline{\text{ERROR}}$  pin was requested by the software even before the failure occurs. In this case, the  $\overline{\text{ERROR}}$  pin is set to high immediately after  $t_{\text{ERROR\_low}}$  expires. This case is not recommended and should be avoided by the application.



**Figure 20-95.  $\overline{\text{ERROR}}$  Pin Timing - Example 5**

### 20.5.5.3 Forcing an Error Condition

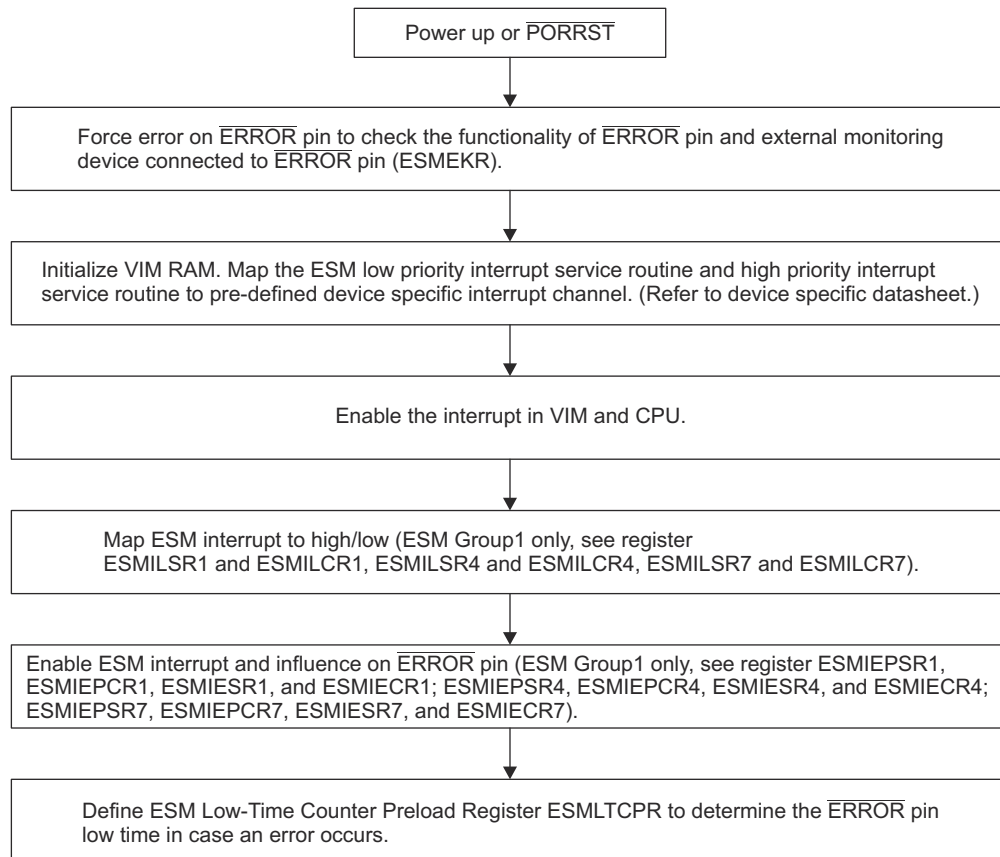
The error response generation mechanism is testable by software by forcing an error condition. This allows testing the  $\overline{\text{ERROR}}$  pin functionality. By writing a dedicated key to the error forcing key register (ESMEKR), the  $\overline{\text{ERROR}}$  pin is set to low for the specified time. The following steps describe how to force an error condition:

1. Check  $\overline{\text{ERROR}}$  Pin Status Register (ESMEPSR). This register must be 1 to switch into the error forcing mode. The ESM module cannot be switched into the error forcing mode if a failure has already been detected in functional mode. The application command to switch to error forcing mode is ignored.
2. Write "1010b" to the error forcing key register (ESMEKR). After that, the  $\overline{\text{ERROR}}$  pin should output low (error force mode). Once the application puts the ESM module in the error forcing mode, the  $\overline{\text{ERROR}}$  pin cannot indicate the normal error functionality. If a failure occurs during this time, it gets still latched and the LTC is reset and stopped. The error output pin is already driven low on account of the error forcing mode. When the ESM is forced back to normal functional mode, the LTC becomes active and forces the  $\overline{\text{ERROR}}$  pin low until the expiration of the LTC.
3. Write "0000" to the error forcing key register (ESMEKR) back to the active normal mode. If there are no errors detected while the ESM module is in the error forcing mode, the  $\overline{\text{ERROR}}$  pin goes high immediately after exiting the error forcing mode.

### 20.5.6 Recommended Programming Procedure

During the initialization stage, the application code should follow the recommendations in [Figure 20-96](#) to initialize the ESM.

Once an error occurs, it can trigger an interrupt,  $\overline{\text{ERROR}}$  pin outputs low depending on the ESM settings. Once the  $\overline{\text{ERROR}}$  pin outputs low, a power on reset or a write of 0x5 to ESMEKR is required to release the ESM back to normal state. The application can read the error status registers (ESMSR1, ESMSR4, ESMSR7, ESMSR2, and ESMSR3) to debug the error. If an  $\overline{\text{RST}}$  is triggered or the error interrupt has been served, the error flag of Group2 should be read from ESMSSR2 because the error flag in ESMSR2 will be cleared by  $\overline{\text{RST}}$ .



**Figure 20-96. ESM Initialization**

### 20.5.7 Main Subsystem ESM Interrupt Map

ESM Group	Define Name	Description
<b>ESM GROUP1</b>		
119	MSS_BUS_SAFETY_MSS2MDO	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
118	MSS_BUS_SAFETY_DMM_SLAVE	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
117	MSS_BUS_SAFETY_DMM_MST	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
116	MSS_BUS_SAFETY_GPADC	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
115	MSS_BUS_SAFETY_RET_RAM	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
114	MSS_BUS_SAFETY_MBOX	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control

ESM Group	Define Name	Description
<b>ESM GROUP1</b>		
112 to 113	RESERVED	RESERVED
111	MSS_BUS_SAFETY_DTHE	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
110	MSS_BUS_SAFETY_HSM_SLAVE	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
108 to 109	RESERVED	RESERVED
107	MSS_BUS_SAFETY_MCRC	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
106	MSS_BUS_SAFETY_QSPI	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
105	MSS_BUS_SAFETY_SEC_TPTC_A1_WR	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
104	MSS_BUS_SAFETY_SEC_TPTC_A1_RD	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
103	MSS_BUS_SAFETY_SEC_TPTC_A0_WR	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
102	MSS_BUS_SAFETY_SEC_TPTC_A0_RD	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
101	MSS_BUS_SAFETY_TPTC_B0_W R	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
100	MSS_BUS_SAFETY_TPTC_A1_W R	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
99	MSS_BUS_SAFETY_TPTC_A0_W R	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
98	MSS_BUS_SAFETY_TPTC_B0_R D	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
97	MSS_BUS_SAFETY_TPTC_A1_R D	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
96	MSS_BUS_SAFETY_TPTC_A0_R D	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
95	MSS_BUS_SAFETY_CPSW	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
94	MSS_BUS_SAFETY_HSM_MST	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
93	MSS_BUS_SAFETY_DAP_RS232	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
92	MSS_BUS_SAFETY_CR5B_SLV	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
91	MSS_BUS_SAFETY_CR5A_SLV	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
87 to 90	RESERVED	RESERVED
86	MSS_MPU_MBOX_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
85	MSS_MPU_MBOX_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
84	MSS_MPU_L2_BANKA_ADDR_ER R	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
83	MSS_MPU_L2_BANKA_PROT_ER R	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
82	MSS_MPU_L2_BANKB_ADDR_ER R	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
81	MSS_MPU_L2_BANKB_PROT_ER R	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
80	MSS_MPU_DTHE_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR

ESM Group	Define Name	Description
<b>ESM GROUP1</b>		
79	MSS_MPU_PCRA_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
78	MSS_MPU_QSPI_ADDR_ERR	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
77	MSS_MPU_CR5A_SLV_ADDR_ER R	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
76	MSS_MPU_CR5B_SLV_ADDR_ER R	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
75	MSS_MPU_HSM_SLV_ADDR_ER R	MPU Addressing Error for MSS_MPU_(X)_ADDR_ERR
74	MSS_MPU_DTHE_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
73	MSS_MPU_PCRA_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
72	MSS_MPU_QSPI_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
71	MSS_MPU_CR5A_SLV_PROT_ER R	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
70	MSS_MPU_CR5B_SLV_PROT_ER R	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
69	MSS_MPU_HSM_SLV_PROT_ERR	MPU Protection Error for MSS_MPU_(X)_ADDR_ERR
68	MSS_CPSW_SERR	Cpsw memories Single bit error pulse
67	MSS_CPSW_UERR	Cpsw memories Double bit error pulse
66	MSS_BUS_SAFETY_SEC_AGG_E RR	Aggregated error for SEC from all Nodes in MSS_SCR
64 to 65	RESERVED	RESERVED
63	ANA_LIMP_MODE	Error signal from analog if the CLK monitor finds the REF CLK to be outside the permissible range of frequency
62	MSS_MCRC_ERR	MCRC Comparision Error
61	MSS_DCCA_ERR	DCCA frequency comparision error
60	MSS_DCCB_ERR	DCCB frequency comparision error
59	MSS_DCCC_ERR	DCCC frequency comparision error
58	MSS_DCCD_ERR	DCCD frequency comparision error
57	MSS_CCCA_ERR	CCCA frequency comparision error
56	MSS_CCCB_ERR	CCCB frequency comparision error
55	MSS_SPIA_SERR	Single Bit correctable error indication for MIBSPI-A multi-buffer
54	MSS_SPIB_SERR	Single Bit correctable error indication for MIBSPI-B multi-buffer
53	MSS_SPIA_UERR	Multi Bit uncorrectable error indication for MIBSPI-A muti-buffer
52	MSS_SPIB_UERR	Multi Bit uncorrectable error indication for MIBSPI-B muti-buffer
51	MSS_MCANA_SERR	Single Bit correctable error indication for MCANA Message Memory
50	MSS_MCANA_UERR	Multi Bit uncorrectable error indication for MCANA Message Memory
49	MSS_MCANA_TS_ERR	MCANA Timestamping Error
48	MSS_MCANB_SERR	Single Bit correctable error indication for MCANB Message Memory
47	MSS_MCANB_UERR	Multi Bit uncorrectable error indication for MCANB Message Memory
46	MSS_MCANB_TS_ERR	MCANB Timestamping Error
45	PAD_NERROR_IN	Nerror from PAD
44	MSS_TCMA_CR5A_SERR	Single Bit correctable error indication for ATCM of CR5A
43	MSS_TCMB1_CR5A_SERR	Single Bit correctable error indication for B1TCM of CR5A
42	MSS_TCMB0_CR5A_SERR	Single Bit correctable error indication for B0TCM of CR5A
41	MSS_TCMA_CR5B_SERR	Single Bit correctable error indication for ATCM of CR5B
40	MSS_TCMB1_CR5B_SERR	Single Bit correctable error indication for B1TCM of CR5B
39	MSS_TCMB0_CR5B_SERR	Single Bit correctable error indication for B0TCM of CR5B

ESM Group	Define Name	Description
<b>ESM GROUP1</b>		
38	MSS_CR5A_ITAG_SERR	Single Bit correctable error indication for Cache ITAG of CR5A
37	MSS_CR5A_IDATA_SERR	Single Bit correctable error indication for Cache IDATA of CR5A
36	MSS_CR5A_DTAG_SERR	Single Bit correctable error indication for Cache DTAG of CR5A
35	MSS_CR5A_DDATA_SERR	Single Bit correctable error indication for Cache DDATA of CR5A
34	MSS_CR5B_ITAG_SERR	Single Bit correctable error indication for Cache ITAG of CR5B
33	MSS_CR5B_IDATA_SERR	Single Bit correctable error indication for Cache IDATA of CR5B
32	MSS_CR5B_DTAG_SERR	Single Bit correctable error indication for Cache DTAG of CR5B
31	MSS_CR5B_DDATA_SERR	Single Bit correctable error indication for Cache DDATA of CR5B
30	MSS_TPCC_A_AGG_ERR	MSS_TPCC_A Aggregated Error Interrupt - TPCC Error - TPCC MPU Error - TPTC Error for all TPTCs connected to TPCC - Read and Write Config Space Access error to TPCC - Read and Write Config Space Access error or all TPTCs connected to TPCC
29	MSS_TPCC_B_AGG_ERR	MSS_TPCC_B Aggregated Error Interrupt - TPCC Error - TPCC MPU Error - TPTC Error for all TPTCs connected to TPCC - Read and Write Config Space Access error to TPCC - Read and Write Config Space Access error or all TPTCs connected to TPCC
28	RESERVED	RESERVED
27	EFUSE_ERR	Reserved for efuse errors
26	MSS_STC_ERR	STC Error indication for MSS Cortex5ss
25	MSS_CCMR5_ST_ERR	CORTEXR5-Sub System Self test error for CCMR5 (comparator module)
24	RESERVED	RESERVED
23	QSPI_WR_ERR	QSPI write error
22	MSS_ECC_AGGR_CR5A_SERR	MSS ECC AGGR for CR5A Memories Correctbale Error- MSS_VIM_CR5A - MSS_CR5A_CACHES (only for injection. Error is sent out through event bus) - MSS_CR5A_TCMs (only for injection. Error is sent out through event bus)
21	RESERVED	RESERVED
20	MSS_ECC_AGGR_CR5B_SERR	MSS ECC AGGR for CR5B Memories Correctbale Error- MSS_VIM_CR5B - MSS_CR5B_CACHES (only for injection. Error is sent out through event bus) - MSS_CR5B_TCMs (only for injection. Error is sent out through event bus)
19	RESERVED	RESERVED
18	MSS_ECC_AGGR_SERR	MSS ECC AGGR Correctbale Error- MSS_L2_BANKA/B - MSS_MBOX - MSS_RETRAM- MSS_GPADC- MSS_TPTC_A0/1 FIFO - MSS_TPTC_B0 FIFO
17	MSS_ECC_AGGR_UERR	MSS ECC AGGR Un-Correctbale Error- MSS_L2_BANKA/B - MSS_MBOX - MSS_RETRAM- MSS_GPADC- MSS_TPTC_A0/1 FIFO - MSS_TPTC_B0 FIFO
14 to 16	RESERVED	RESERVED
13	DSS_ESM_LO	ESM IRQ from Gem
12	DSS_ESM_HI	ESM FIQ from Gem
1	HSM_ESM_LO	ESM IRQ from HSM
0	HSM_ESM_HI	ESM FIQ fromHSM
<b>ESM GROUP2</b>		
24	MSS_BUS_SAFETY_CR5A_MST_RD	MSS Bus Safety Error for the Node MSSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
23	MSS_BUS_SAFETY_CR5B_MST_RD	MSS Bus Safety Error for the Node MSSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control
22	MSS_BUS_SAFETY_CR5A_MST_WR	MSS Bus Safety Error for the Node MSSSS_BUS_SAFETY_(X) - ECC Uncorrectable Data Error - Compare Error on Control

ESM Group	Define Name	Description
<b>ESM GROUP1</b>		
21	MSS_BUS_SAFETY_CR5B_MST_WR	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY(X) - ECC Uncorrectable Data Error - Compare Error on Control
20	MSS_BUS_SAFETY_L2_BANKA	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY(X) - ECC Uncorrectable Data Error - Compare Error on Control
19	MSS_BUS_SAFETY_L2_BANKB	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY(X) - ECC Uncorrectable Data Error - Compare Error on Control
18	MSS_BUS_SAFETY_PCRA	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY(X) - ECC Uncorrectable Data Error - Compare Error on Control
17	MSS_BUS_SAFETY_PCRB	MSS Bus Safety Error for the Node MSSS_BUS_SAFETY(X) - ECC Uncorrectable Data Error - Compare Error on Control
16	MSS_ECC_AGGR_CR5A_UERR	MSS ECC AGGR for CR5A Memories Un-Correctable Error- MSS_VIM_CR5A - MSS_CR5A_CACHES (only for injection. Error is sent out through event bus) - MSS_CR5A_TCMs (only for injection. Error is sent out through event bus)
15	MSS_ECC_AGGR_CR5B_UERR	MSS ECC AGGR for CR5A Memories Un-Correctable Error- MSS_VIM_CR5A - MSS_CR5A_CACHES (only for injection. Error is sent out through event bus) - MSS_CR5A_TCMs (only for injection. Error is sent out through event bus)
14	MSS_L2_BANKA_ECC_UERR	MSS_L2_BANKA Uncorrectable ECC Error
13	MSS_L2_BANKB_ECC_UERR	MSS_L2_BANKB Uncorrectable ECC Error
12	VIM_LOCK_ERR	MSS_VIM lock step compare error
11	MSS_WDT_NMI	MSS Watch dog timer non maskable irq
10	MSS_CR5A_LIVELOCK	MSS_CR5A in live lock due to fatal errors
9	MSS_CR5B_LIVELOCK	MSS_CR5B in live lock due to fatal errors
8	MSS_TCMB1_CR5B_PARITY_ERR	Parity Error on Control signals for B1TCM of CR5B
7	MSS_TCMB0_CR5B_PARITY_ERR	Parity Error on Control signals for B0TCM of CR5B
6	MSS_TCMA_CR5B_PARITY_ERR	Parity Error on Control signals for ATCM of CR5B
5	MSS_TCMB1_CR5A_PARITY_ERR	Parity Error on Control signals for B1TCM of CR5A
4	MSS_TCMB0_CR5A_PARITY_ERR	Parity Error on Control signals for B0TCM of CR5A
3	MSS_TCMA_CR5A_PARITY_ERR	Parity Error on Control signals for ATCM of CR5A
2	MSS_CCMR5_ERR	Lock step Comparison Error from CCMR5
1	DSS_ESM_HI	ESM FIQ from DSP
0	HSM_ESM_HI	ESM FIQ from HSM

ESM GROUP3	Define Name	Description
31	RESERVED	RESERVED
30	RESERVED	RESERVED
29	MSS_CR5B_DDATA_UERR	Dcache data multibit error from CR5B
28	RESERVED	RESERVED
27	MSS_CR5B_DTAG_UERR	Dcache tag multibit error from CR5B
26	RESERVED	RESERVED
25	RESERVED	RESERVED
24	RESERVED	RESERVED
23	RESERVED	RESERVED
22	RESERVED	RESERVED
21	MSS_CR5A_DDATA_UERR	Dcache data multibit error from CR5A
20	RESERVED	RESERVED
19	MSS_CR5A_DTAG_UERR	Dcache tag multibit error from CR5A
18	RESERVED	RESERVED

ESM GROUP3	Define Name	Description
17	RESERVED	RESERVED
16	RESERVED	RESERVED
15	RESERVED	RESERVED
14	RESERVED	RESERVED
13	MSS_TCMA_CR5B_UERR	Multi Bit Error in ATCM of CR5B
12	RESERVED	RESERVED
11	MSS_TCMB1_CR5B_UERR	Multi Bit Error in B1TCM of CR5B
10	RESERVED	RESERVED
9	MSS_TCMB0_CR5B_UERR	Multi Bit Error in B0TCM of CR5B
8	RESERVED	RESERVED
7	MSS_TCMA_CR5A_UERR	Multi Bit Error in ATCM of CR5A
6	RESERVED	RESERVED
5	MSS_TCMB1_CR5A_UERR	Multi Bit Error in B1TCM of CR5A
4	RESERVED	RESERVED
3	MSS_TCMB0_CR5A_UERR	Multi Bit Error in B0TCM of CR5A
2	RESERVED	RESERVED
1	EFUSE_AUTOLOAD_ERR	Reserved for efuse autoloader error
0	RESERVED	RESERVED



## 20.5.8 MSS\_ESM Registers

Table 20-119 lists the memory-mapped registers for the MSS\_ESM. To determine the base address, refer to the device-specific memory map. The address locations not listed are reserved.

**Table 20-119. MSS\_ESM Registers**

Offset	Acronym	Register Name	Section
0h	ESMIEPSR1	ESM Enable ERROR Pin Action/Response Register 1	<a href="#">Section 20.5.8.1</a>
4h	ESMIEPCR1	ESM Disable ERROR Pin Action/Response Register 1	<a href="#">Section 20.5.8.2</a>
8h	ESMIESR1	ESM Interrupt Enable Set/Status Register 1	<a href="#">Section 20.5.8.3</a>
Ch	ESMIECR1	ESM Interrupt Enable Clear/Status Register 1	<a href="#">Section 20.5.8.4</a>
10h	ESMILSR1	Interrupt Level Set/Status Register 1	<a href="#">Section 20.5.8.5</a>
14h	ESMILCR1	Interrupt Level Clear/Status Register 1	<a href="#">Section 20.5.8.6</a>
18h	ESMSR1	ESM Status Register 1	<a href="#">Section 20.5.8.7</a>
1Ch	ESMSR2	ESM Status Register 2	<a href="#">Section 20.5.8.8</a>
20h	ESMSR3	ESM Status Register 3	<a href="#">Section 20.5.8.9</a>
24h	ESMEPSR	ESM ERROR Pin Status Register	<a href="#">Section 20.5.8.10</a>
28h	ESMIOFFHR	ESM Interrupt Offset High Register	<a href="#">Section 20.5.8.11</a>
2Ch	ESMIOFFLR	ESM Interrupt Offset Low Register	<a href="#">Section 20.5.8.12</a>
30h	ESMLTCR	ESM Low-Time Counter Register	<a href="#">Section 20.5.8.13</a>
34h	ESMLTCPR	ESM Low-Time Counter Preload Register	<a href="#">Section 20.5.8.14</a>
38h	ESMEKR	ESM Error Key Register	<a href="#">Section 20.5.8.15</a>
3Ch	ESMSSR2	ESM Status Shadow Register 2	<a href="#">Section 20.5.8.16</a>
40h	ESMIEPSR4	ESM Enable ERROR Pin Action/Response Register 4	<a href="#">Section 20.5.8.17</a>
44h	ESMIEPCR4	ESM Disable ERROR Pin Action/Response Register 4	<a href="#">Section 20.5.8.18</a>
48h	ESMIESR4	ESM Interrupt Enable Set/Status Register 4	<a href="#">Section 20.5.8.19</a>
4Ch	ESMIECR4	ESM Interrupt Enable Clear/Status Register 4	<a href="#">Section 20.5.8.20</a>
50h	ESMILSR4	Interrupt Level Set/Status Register 4	<a href="#">Section 20.5.8.21</a>
54h	ESMILCR4	Interrupt Level Clear/Status Register 4	<a href="#">Section 20.5.8.22</a>
58h	ESMSR4	ESM Status Register 4	<a href="#">Section 20.5.8.23</a>
80h	ESMIEPSR7	ESM Enable ERROR Pin Action/Response Register 7	<a href="#">Section 20.5.8.24</a>
84h	ESMIEPCR7	ESM Disable ERROR Pin Action/Response Register 7	<a href="#">Section 20.5.8.25</a>
88h	ESMIESR7	ESM Interrupt Enable Set/Status Register 7	<a href="#">Section 20.5.8.26</a>
8Ch	ESMIECR7	ESM Interrupt Enable Clear/Status Register 7	<a href="#">Section 20.5.8.27</a>
90h	ESMILSR7	Interrupt Level Set/Status Register 7	<a href="#">Section 20.5.8.28</a>
94h	ESMILCR7	Interrupt Level Clear/Status Register 7	<a href="#">Section 20.5.8.29</a>
98h	ESMSR7	ESM Status Register 7	<a href="#">Section 20.5.8.30</a>

Complex bit access types are encoded to fit into small table cells. Table 20-120 shows the codes that are used for access types in this section.

**Table 20-120. MSS\_ESM Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		

**Table 20-120. MSS\_ESM Access Type Codes  
(continued)**

Access Type	Code	Description
$-n$		Value after reset or the default value

### 20.5.8.1 ESMIEPSR1 Register (Offset = 0h) [reset = 0h]

ESMIEPSR1 is shown in [Figure 20-97](#) and described in [Table 20-121](#).

Return to [Summary Table](#).

ESM Enable ERROR Pin Action/Response Register 1

**Figure 20-97. ESMIEPSR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPSET																															
R/W-0h																															

**Table 20-121. ESMIEPSR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IEPSET	R/W	0h	Enable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding clear bit in the ESMIEPCR1 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Enables failure influence on ERROR pin and sets the corresponding clear bit in the ESMIEPCR1 register.

### 20.5.8.2 ESMIEPCR1 Register (Offset = 4h) [reset = 0h]

ESMIEPCR1 is shown in [Figure 20-98](#) and described in [Table 20-122](#).

Return to [Summary Table](#).

ESM Disable ERROR Pin Action/Response Register 1

**Figure 20-98. ESMIEPCR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPCLR																															
R/W-0h																															

**Table 20-122. ESMIEPCR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IEPCLR	R/W	0h	Disable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding set bit in the ESMIEPSR1 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Disables failure influence on ERROR pin and clears the corresponding set bit in the ESMIEPSR1 register.

### 20.5.8.3 ESMIESR1 Register (Offset = 8h) [reset = 0h]

ESMIESR1 is shown in [Figure 20-99](#) and described in [Table 20-123](#).

Return to [Summary Table](#).

ESM Interrupt Enable Set/Status Register 1

**Figure 20-99. ESMIESR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSET																															
R/W-0h																															

**Table 20-123. ESMIESR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTENSET	R/W	0h	Set interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding clear bit in the ESMIECR1 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR1 register.

#### 20.5.8.4 ESMIECR1 Register (Offset = Ch) [reset = 0h]

ESMIECR1 is shown in [Figure 20-100](#) and described in [Table 20-124](#).

Return to [Summary Table](#).

ESM Interrupt Enable Clear/Status Register 1

**Figure 20-100. ESMIECR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENCLR																															
R/W-0h																															

**Table 20-124. ESMIECR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTENCLR	R/W	0h	Clear Interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding set bit in the ESMIESR1 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR1 register.

### 20.5.8.5 ESMILSR1 Register (Offset = 10h) [reset = 0h]

ESMILSR1 is shown in [Figure 20-101](#) and described in [Table 20-125](#).

Return to [Summary Table](#).

Interrupt Level Set/Status Register 1

**Figure 20-101. ESMILSR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLSET																															
R/W-0h																															

**Table 20-125. ESMILSR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTLVLSET	R/W	0h	Set Interrupt Priority Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding clear bit in the ESMILCR1 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to high level interrupt line and sets the corresponding clear bit in the ESMILCR1 register.

### 20.5.8.6 ESMILCR1 Register (Offset = 14h) [reset = 0h]

ESMILCR1 is shown in [Figure 20-102](#) and described in [Table 20-126](#).

Return to [Summary Table](#).

Interrupt Level Clear/Status Register 1

**Figure 20-102. ESMILCR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLCR																															
R/W-0h																															

**Table 20-126. ESMILCR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTLVLCR	R/W	0h	Clear Interrupt Priority. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding set bit in the ESMILSR1 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to low level interrupt line and clears the corresponding set bit in the ESMILSR1 register.



### 20.5.8.7 ESMSR1 Register (Offset = 18h) [reset = 0h]

ESMSR1 is shown in [Figure 20-103](#) and described in [Table 20-127](#).

Return to [Summary Table](#).

ESM Status Register 1

**Figure 20-103. ESMSR1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

**Table 20-127. ESMSR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. Note: After nRST, if one of these flags are set and the corresponding interrupt are enabled, the interrupt service routine will be called.

### 20.5.8.8 ESMSR2 Register (Offset = 1Ch) [reset = 0h]

ESMSR2 is shown in [Figure 20-104](#) and described in [Table 20-128](#).

Return to [Summary Table](#).

ESM Status Register 2

**Figure 20-104. ESMSR2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

**Table 20-128. ESMSR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. ESMSSR2 is not impacted by this action. Note: In normal operation the flag gets cleared when reading the appropriate vector in the ESMIOFFHR offset register. Reading ESMIOFFHR will not clear the ESMSR1 and the shadow register ESMSSR2.

### 20.5.8.9 ESMSR3 Register (Offset = 20h) [reset = 0h]

ESMSR3 is shown in [Figure 20-105](#) and described in [Table 20-129](#).

Return to [Summary Table](#).

ESM Status Register 3

**Figure 20-105. ESMSR3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

**Table 20-129. ESMSR3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred. Write: Leaves the bit unchanged. 1 Read: Error occurred. Write: Clears the bit.

### 20.5.8.10 ESMEPSR Register (Offset = 24h) [reset = 0h]

ESMEPSR is shown in [Figure 20-106](#) and described in [Table 20-130](#).

Return to [Summary Table](#).

ESM ERROR Pin Status Register

**Figure 20-106. ESMEPSR Register**

31	30	29	28	27	26	25	24
RESERVED							
R/W-0h							
23	22	21	20	19	18	17	16
RESERVED							
R/W-0h							
15	14	13	12	11	10	9	8
RESERVED							
R/W-0h							
7	6	5	4	3	2	1	0
RESERVED							EPSF
R/W-0h							R/W-0h

**Table 20-130. ESMEPSR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
0	EPSF	R/W	0h	ERROR Pin Status Flag. Provides status information for the ERROR Pin. Read/Write in User and Privileged mode. 0 Read: ERROR Pin is low (active) if any error has occurred. Write: Writes have no effect. 1 Read: ERROR Pin is high if no error has occurred. Write: Writes have no effect. Note: This flag will be set to 1 after PORRST. The value will be unchanged after nRST. The ERROR pin status remains un-changed during after nRST.

### 20.5.8.11 ESMIOFFHR Register (Offset = 28h) [reset = 0h]

ESMIOFFHR is shown in [Figure 20-107](#) and described in [Table 20-131](#).

Return to [Summary Table](#).

ESM Interrupt Offset High Register

**Figure 20-107. ESMIOFFHR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								INTOFFH							
R/W-0h																								R/W-0h							

**Table 20-131. ESMIOFFHR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
6-0	INTOFFH	R/W	0h	Offset High Level Interrupt. This vector gives the channel number of the highest pending interrupt request for the high level interrupt line. Interrupts of error Group2 have higher priority than interrupts of error Group1. Inside a group, channel 0 has highest priority and channel 31 has lowest priority. User and privileged mode (read): Returns number of pending interrupt with the highest priority for the high level interrupt line. 0 No pending interrupt. 1h Interrupt pending for channel 0, error Group1. ... 20h Interrupt pending for channel 31, error Group1. 21h Interrupt pending for channel 0, error Group2. ... 40h Interrupt pending for channel 31, error Group2. 41h Interrupt pending for channel 32, error Group1. ... 60h Interrupt pending for channel 63, error Group1. Note: Reading the interrupt vector will clear the corresponding flag in the ESMSR2 register; will not clear ESMSR1 and ESMSR2 and the offset register gets updated. User and privileged mode (write): Writes have no effect.

### 20.5.8.12 ESMIOFFLR Register (Offset = 2Ch) [reset = 0h]

ESMIOFFLR is shown in [Figure 20-108](#) and described in [Table 20-132](#).

Return to [Summary Table](#).

ESM Interrupt Offset Low Register

**Figure 20-108. ESMIOFFLR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								INTOFFL							
R/W-0h																								R/W-0h							

**Table 20-132. ESMIOFFLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-7	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
6-0	INTOFFL	R/W	0h	Offset Low Level Interrupt. This vector gives the channel number of the highest pending interrupt request for the low level interrupt line. Inside a group, channel 0 has highest priority and channel 31 has lowest priority. User and privileged mode (read): Returns number of pending interrupt with the highest priority for the low level interrupt line. 0 No pending interrupt. 1h Interrupt pending for channel 0, error Group1. ... 20h Interrupt pending for channel 31, error Group1. 21h Interrupt pending for channel 32, error Group1. ... 60h Interrupt pending for channel 63, error Group1. Note: Reading the interrupt vector will not clear the corresponding flag in the ESMSR1 register. Group2 interrupts are fixed to the high level interrupt line only. User and privileged mode (write): Writes have no effect.

### 20.5.8.13 ESMLTCR Register (Offset = 30h) [reset = 0h]

ESMLTCR is shown in [Figure 20-109](#) and described in [Table 20-133](#).

Return to [Summary Table](#).

ESM Low-Time Counter Register

**Figure 20-109. ESMLTCR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LTCP															
R/W-0h																R/W-0h															

**Table 20-133. ESMLTCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
15-0	LTCP	R/W	0h	ERROR Pin Low-Time Counter 16bit pre-loadable down-counter to control low-time of ERROR pin. The low-time counter is triggered by the peripheral clock (VCLK). Note: Low time counter is set to the default preload value of the ESMLTCPR in the following cases: 1. Reset (power on reset or warm reset) 2. An error occurs 3. User forces an error

#### 20.5.8.14 ESMLTCPR Register (Offset = 34h) [reset = 0h]

ESMLTCPR is shown in [Figure 20-110](#) and described in [Table 20-134](#).

Return to [Summary Table](#).

ESM Low-Time Counter Preload Register

**Figure 20-110. ESMLTCPR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LTCP															
R/W-0h																R/W-0h															

**Table 20-134. ESMLTCPR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
15-0	LTCP	R/W	0h	ERROR Pin Low-Time Counter Pre-load Value 16bit pre-load value for the ERROR pin low-time counter. Note: Only LTCP.15 and LTCP.14 are configurable (privileged mode write).



### 20.5.8.15 ESMEKR Register (Offset = 38h) [reset = 0h]

ESMEKR is shown in [Figure 20-111](#) and described in [Table 20-135](#).

Return to [Summary Table](#).

ESM Error Key Register

**Figure 20-111. ESMEKR Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EKEY			
R/W-0h																												R/W-0h			

**Table 20-135. ESMEKR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-4	RESERVED	R/W	0h	Read returns 0. Writes have no effect.
3-0	EKEY	R/W	0h	Error Key. The key to reset the ERROR pin or to force an error on the ERROR pin. User and privileged mode (read): Returns current value of the EKEY. Privileged mode (write): 0 Activates normal mode (recommended default mode). Ah Forces error on ERROR pin. 5h The ERROR pin set to high when the low time counter (LTC) has completed; then the EKEY bit will switch back to normal mode (EKEY = 0000) All other values Activates normal mode.

### 20.5.8.16 ESMSSR2 Register (Offset = 3Ch) [reset = 0h]

ESMSSR2 is shown in [Figure 20-112](#) and described in [Table 20-136](#).

Return to [Summary Table](#).

ESM Status Shadow Register 2

**Figure 20-112. ESMSSR2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

**Table 20-136. ESMSSR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Shadow register for status information on pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred. Write: Leaves the bit unchanged. 1 Read: Error occurred. Write: Clears the bit. ESMSSR2 is not impacted by this action. Note: Errors are stored until they are cleared by the software or at power-on reset (PORRST).

### 20.5.8.17 ESMIEPSR4 Register (Offset = 40h) [reset = 0h]

ESMIEPSR4 is shown in [Figure 20-113](#) and described in [Table 20-137](#).

Return to [Summary Table](#).

ESM Enable ERROR Pin Action/Response Register 4

**Figure 20-113. ESMIEPSR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPSET																															
R/W-0h																															

**Table 20-137. ESMIEPSR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IEPSET	R/W	0h	Enable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding clear bit in the ESMIEPCR4 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Enables failure influence on ERROR pin and sets the corresponding clear bit in the ESMIEPCR4 register.

### 20.5.8.18 ESMIEPCR4 Register (Offset = 44h) [reset = 0h]

ESMIEPCR4 is shown in [Figure 20-114](#) and described in [Table 20-138](#).

Return to [Summary Table](#).

ESM Disable ERROR Pin Action/Response Register 4

**Figure 20-114. ESMIEPCR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPCLR																															
R/W-0h																															

**Table 20-138. ESMIEPCR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IEPCLR	R/W	0h	Disable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding set bit in the ESMIEPSR4 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Disables failure influence on ERROR pin and clears the corresponding set bit in the ESMIEPSR4 register.

### 20.5.8.19 ESMIESR4 Register (Offset = 48h) [reset = 0h]

ESMIESR4 is shown in [Figure 20-115](#) and described in [Table 20-139](#).

Return to [Summary Table](#).

ESM Interrupt Enable Set/Status Register 4

**Figure 20-115. ESMIESR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSET																															
R/W-0h																															

**Table 20-139. ESMIESR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTENSET	R/W	0h	Set interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding clear bit in the ESMIECR4 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR4 register.

### 20.5.8.20 ESMIECR4 Register (Offset = 4Ch) [reset = 0h]

ESMIECR4 is shown in [Figure 20-116](#) and described in [Table 20-140](#).

Return to [Summary Table](#).

ESM Interrupt Enable Clear/Status Register 4

**Figure 20-116. ESMIECR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENCLR																															
R/W-0h																															

**Table 20-140. ESMIECR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTENCLR	R/W	0h	Clear Interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding set bit in the ESMIESR4 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR4 register.

### 20.5.8.21 ESMILSR4 Register (Offset = 50h) [reset = 0h]

ESMILSR4 is shown in [Figure 20-117](#) and described in [Table 20-141](#).

Return to [Summary Table](#).

Interrupt Level Set/Status Register 4

**Figure 20-117. ESMILSR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLSET																															
R/W-0h																															

**Table 20-141. ESMILSR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTLVLSET	R/W	0h	Set Interrupt Priority Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding clear bit in the ESMILCR4 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to high level interrupt line and sets the corresponding clear bit in the ESMILCR4 register.

### 20.5.8.22 ESMILCR4 Register (Offset = 54h) [reset = 0h]

ESMILCR4 is shown in [Figure 20-118](#) and described in [Table 20-142](#).

Return to [Summary Table](#).

Interrupt Level Clear/Status Register 4

**Figure 20-118. ESMILCR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLCR																															
R/W-0h																															

**Table 20-142. ESMILCR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTLVLCR	R/W	0h	Clear Interrupt Priority. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding set bit in the ESMILSR4 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to low level interrupt line and clears the corresponding set bit in the ESMILSR4 register.



### 20.5.8.23 ESMSR4 Register (Offset = 58h) [reset = 0h]

ESMSR4 is shown in [Figure 20-119](#) and described in [Table 20-143](#).

Return to [Summary Table](#).

ESM Status Register 4

**Figure 20-119. ESMSR4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

**Table 20-143. ESMSR4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. Note: After nRST, if one of these flags are set and the corresponding interrupt are enabled, the interrupt service routine will be called.

### 20.5.8.24 ESMIEPSR7 Register (Offset = 80h) [reset = 0h]

ESMIEPSR7 is shown in [Figure 20-120](#) and described in [Table 20-144](#).

Return to [Summary Table](#).

ESM Enable ERROR Pin Action/Response Register 7

**Figure 20-120. ESMIEPSR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPSET																															
R/W-0h																															

**Table 20-144. ESMIEPSR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IEPSET	R/W	0h	Enable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding clear bit in the ESMIEPCR7 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Enables failure influence on ERROR pin and sets the corresponding clear bit in the ESMIEPCR7 register.

### 20.5.8.25 ESMIEPCR7 Register (Offset = 84h) [reset = 0h]

ESMIEPCR7 is shown in [Figure 20-121](#) and described in [Table 20-145](#).

Return to [Summary Table](#).

ESM Disable ERROR Pin Action/Response Register 7

**Figure 20-121. ESMIEPCR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IEPCLR																															
R/W-0h																															

**Table 20-145. ESMIEPCR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	IEPCLR	R/W	0h	Disable ERROR Pin Action/Response on Group 1. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Failure on channel x has no influence on ERROR pin. Write: Leaves the bit and the corresponding set bit in the ESMIEPSR7 register unchanged. 1 Read: Failure on channel x has influence on ERROR pin. Write: Disables failure influence on ERROR pin and clears the corresponding set bit in the ESMIEPSR7 register.

### 20.5.8.26 ESMIESR7 Register (Offset = 88h) [reset = 0h]

ESMIESR7 is shown in [Figure 20-122](#) and described in [Table 20-146](#).

Return to [Summary Table](#).

ESM Interrupt Enable Set/Status Register 7

**Figure 20-122. ESMIESR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENSET																															
R/W-0h																															

**Table 20-146. ESMIESR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTENSET	R/W	0h	Set interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding clear bit in the ESMIECR7 register unchanged. 1 Read: Interrupt is enabled. Write: Enables interrupt and sets the corresponding clear bit in the ESMIECR7 register.

### 20.5.8.27 ESMIECR7 Register (Offset = 8Ch) [reset = 0h]

ESMIECR7 is shown in [Figure 20-123](#) and described in [Table 20-147](#).

Return to [Summary Table](#).

ESM Interrupt Enable Clear/Status Register 7

**Figure 20-123. ESMIECR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTENCLR																															
R/W-0h																															

**Table 20-147. ESMIECR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTENCLR	R/W	0h	Clear Interrupt Enable Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt is disabled. Write: Leaves the bit and the corresponding set bit in the ESMIESR7 register unchanged. 1 Read: Interrupt is enabled. Write: Disables interrupt and clears the corresponding set bit in the ESMIESR7 register.

### 20.5.8.28 ESMILSR7 Register (Offset = 90h) [reset = 0h]

ESMILSR7 is shown in [Figure 20-124](#) and described in [Table 20-148](#).

Return to [Summary Table](#).

Interrupt Level Set/Status Register 7

**Figure 20-124. ESMILSR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLSET																															
R/W-0h																															

**Table 20-148. ESMILSR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTLVLSET	R/W	0h	Set Interrupt Priority Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding clear bit in the ESMILCR7 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to high level interrupt line and sets the corresponding clear bit in the ESMILCR7 register.

### 20.5.8.29 ESMILCR7 Register (Offset = 94h) [reset = 0h]

ESMILCR7 is shown in [Figure 20-125](#) and described in [Table 20-149](#).

Return to [Summary Table](#).

Interrupt Level Clear/Status Register 7

**Figure 20-125. ESMILCR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLVLCR																															
R/W-0h																															

**Table 20-149. ESMILCR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	INTLVLCR	R/W	0h	Clear Interrupt Priority. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: Interrupt of channel x is mapped to low level interrupt line. Write: Leaves the bit and the corresponding set bit in the ESMILSR7 register unchanged. 1 Read: Interrupt of channel x is mapped to high level interrupt line. Write: Maps interrupt of channel x to low level interrupt line and clears the corresponding set bit in the ESMILSR7 register.

### 20.5.8.30 ESMSR7 Register (Offset = 98h) [reset = 0h]

ESMSR7 is shown in [Figure 20-126](#) and described in [Table 20-150](#).

Return to [Summary Table](#).

ESM Status Register 7

**Figure 20-126. ESMSR7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ESF																															
R/W-0h																															

**Table 20-150. ESMSR7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ESF	R/W	0h	Error Status Flag. Provides status information on a pending error. Read in User and Privileged mode. Write in Privileged mode only. 0 Read: No error occurred; no interrupt is pending. Write: Leaves the bit unchanged. 1 Read: Error occurred; interrupt is pending. Write: Clears the bit. Note: After nRST, if one of these flags are set and the corresponding interrupt are enabled, the interrupt service routine will be called.

## 20.6 Cyclic Redundancy Check (CRC)

This section describes the cyclic redundancy check (CRC) controller module. Presently two CRC modules have been instantiated in the one for MSS and another for DSS.

### 20.6.1 Overview

The CRC controller is a module that is used to perform CRC (Cyclic Redundancy Check) to verify the integrity of memory system. A signature representing the contents of the memory is obtained when the contents of the memory are read into CRC controller. The responsibility of CRC controller is to calculate the signature for a set of data and then compare the calculated signature value against a pre-determined good signature value. CRC controller supports two channels to perform CRC calculation on multiple memories in parallel and can be used on any memory system.

### 20.6.2 Features

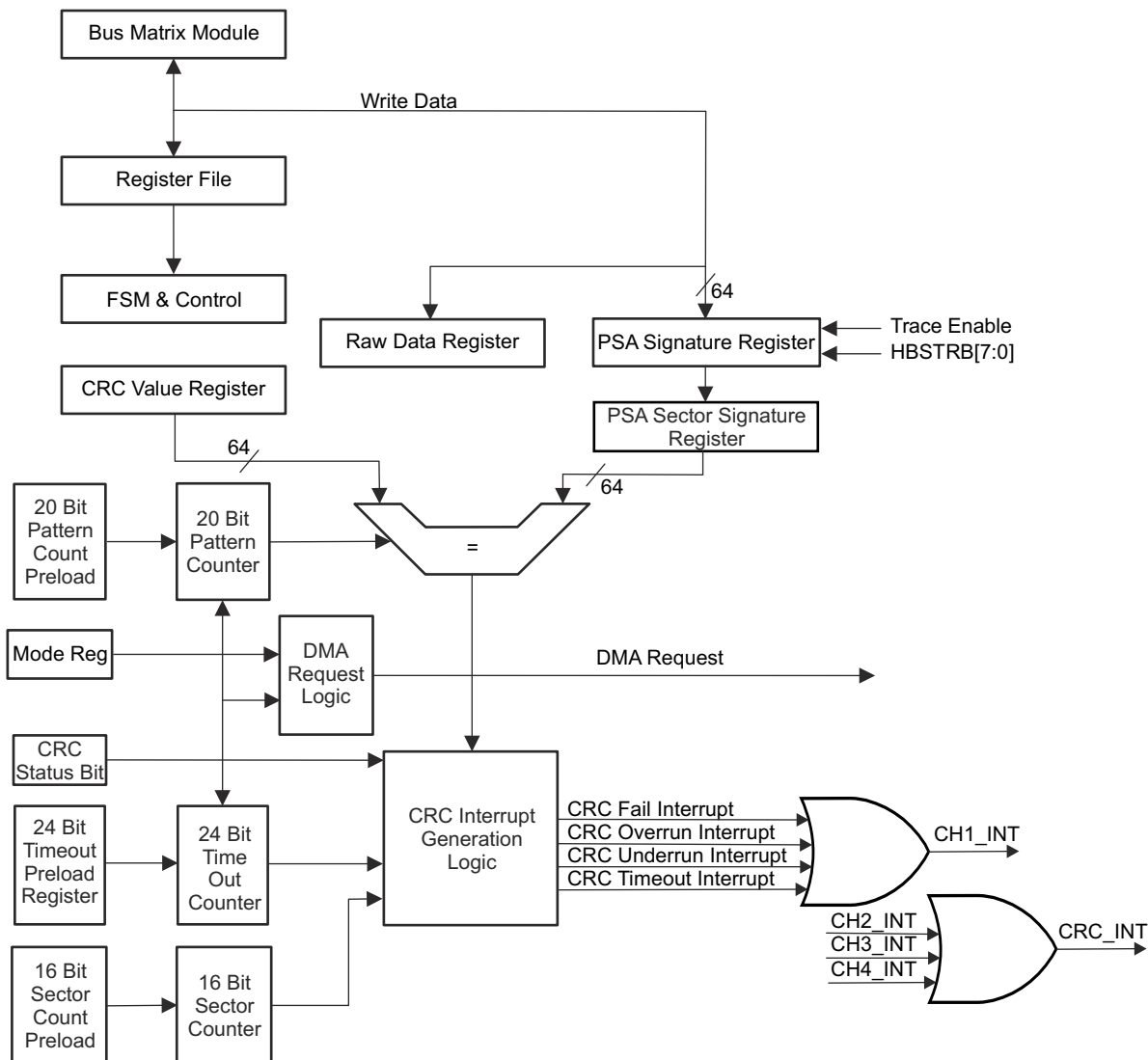
The CRC controller offers:

- Two channels to perform background signature verification on any memory sub-system.
- Data compression on 8, 16, 32, and 64-bit data size.
- Maximum-length PSA (Parallel Signature Analysis) register constructed based on 64-bit primitive polynomial.
- Each channel has a CRC Value Register that contains the pre-determined CRC value.
- Use timed base event trigger from timer to initiate DMA data transfer.
- Programmable 20-bit pattern counter per channel to count the number of data patterns for compression.
- Three modes of operation. Auto, Semi-CPU and Full-CPU.
- For each channel, CRC can be performed either by CRC Controller or by CPU.
- Automatically perform signature verification without CPU intervention in AUTO mode.
- Generate interrupt to CPU in Semi-CPU mode to allow CPU to perform signature verification itself.
- Generate CRC fail interrupt in AUTO mode if signature verification fails.
- Generate Timeout interrupt if CRC is not performed within the time limit.
- Generate DMA request per channel to initiate CRC value transfer.

### 20.6.3 Block Diagram

[Figure 20-127](#) shows a block diagram of the CRC controller.





**Figure 20-127. CRC Controller Block Diagram For One Channel**

## **20.6.4 Module Operation**

### **20.6.4.1 General Operation**

There are two channels in CRC controller, and for each channel there is a memory-mapped PSA (Parallel Signature Analysis) Signature Register and a memory-mapped CRC (Cyclic Redundancy Check) Value Register. A memory can be organized into multiple sectors with each sector consisting of multiple data patterns. A data pattern can be 8-, 16-, 32-, or 64-bit data. CRC module performs the signature calculation and compares the signature to a pre-determined value. The PSA Signature Register compresses an incoming data pattern into a signature when it is written. When one sector of data patterns are written into PSA Signature Register, a final signature corresponding to the sector is obtained. CRC Value Register stores the pre-determined signature corresponding to one sector of data patterns. The calculated signature and the pre-determined signature are then compared to each other for signature verification. To minimize CPU's involvement, data patterns transfer can be carried out at the background of CPU using DMA controller. DMA is setup to transfer data from memory from which the contents to be verified to the memory mapped PSA Signature Register. When DMA transfers data to the memory mapped PSA Signature Register, a signature is generated. A programmable 20-bit data pattern counter is used for each channel to define the number of data patterns to calculate for each sector. Signature verification can be performed automatically by CRC controller in AUTO mode or by CPU itself in Semi-CPU or Full-CPU mode. In AUTO mode, a self sustained CRC signature calculation can be achieved without any CPU intervention.

### **20.6.4.2 CRC Modes of Operation**

CRC Controller can operate in AUTO, Semi-CPU, and Full-CPU modes.

#### **20.6.4.2.1 AUTO Mode**

In AUTO mode, CRC Controller in conjunction with DMA controller can perform CRC without CPU intervention. A sustained transfer of data to both the PSA Signature Register and CRC Value Register are performed in the background of CPU. When a mismatch is detected, an interrupt is generated to CPU. A 16-bit current sector ID register is provided to identify which sector causes a CRC failure.

#### **20.6.4.2.2 Semi-CPU Mode**

In Semi-CPU mode, DMA controller is also utilized to perform data patterns transfer to PSA Signature Register. Instead of performing signature verification automatically, the CRC controller generates an compression complete interrupt to CPU after each sector is compressed. Upon responding to the interrupt the CPU performs the signature verification by reading the calculated signature stored at the PSA Sector Signature Register, and compares it to a pre-determined CRC value.

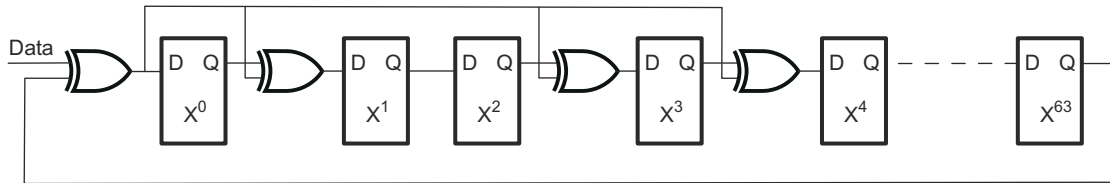
#### **20.6.4.2.3 Full CPU Mode**

In Full-CPU mode, the CPU does the data patterns transfer and signature verification all by itself. When CPU has enough throughput, it can perform data patterns transfer by reading data from the memory system to the PSA Signature Register. After certain number of data patterns are compressed, the CPU can read from the PSA Signature Register and compare the calculated signature to the pre-determined CRC signature value. In Full-CPU mode, neither interrupt nor DMA request is generated. All counters are also disabled.

### 20.6.4.3 PSA Signature Register

The 64-bit PSA Signature Register is based on the primitive polynomial (as in the following equation) to produce the maximum length LFSR (Linear Feedback Shift Register), as shown in [Figure 20-128](#).

$$f(x) = x^{64} + x^4 + x^3 + x + 1 \quad (9)$$



**Figure 20-128. Linear Feedback Shift Register (LFSR)**

The serial implementation of LFSF has a limitation that, it requires 'n' clock cycles to calculate the CRC values for an 'n' bit data stream. The idea is to produce the same CRC value operating on a multi-bit data stream, as would occur if the CRC were computed one bit at a time over the whole data stream. The algorithm involves looping to simulate the shifting, and concatenating strings to build the equations after 'n' shift.

The parallel CRC calculation based on the polynomial can be illustrated in the following HDL code:

```
for i in 63 to 0 loop
  NEXT_CRC_VAL(0) := CRC_VAL(63) xor DATA(i);
  for j in 1 to 63 loop
    case j is
      when 1|3|4 =>
        NEXT_CRC_VAL(j) :=
          CRC_VAL(j - 1) xor CRC_VAL(63) xor DATA(i);
      when others =>
        NEXT_CRC_VAL(j) := CRC_VAL(j - 1);
    end case;
  end loop;
  CRC_VAL := NEXT_CRC_VAL;
end loop;
```

#### Note

- 1) The inner loop is to calculate the next value of each shift register bit after one cycle
- 2) The outer loop is to simulate 64 cycles of shifting. The equation for each shift register bit is thus built before it is compressed into the shift register.
- 3) MSB of the DATA is shifted in first

There is one PSA Signature Register per CRC channel. PSA Signature Register can be both read and written. When it is written, it can either compress the data or just capture the data depending on the state of CHx\_MODE bits. If CHx\_MODE=Data Capture, a seed value can be planted in the PSA Signature Register without compression. Other modes other than Data Capture will result with the data compressed by PSA Signature Register when it is written. Each channel can be planted with different seed value before compression starts. When PSA Signature Register is read, it gives the calculated signature.

CRC Controller should be used in conjunction with the on chip DMA controller to produce optimal system performance. The incoming data pattern to PSA Signature Register is typically initiated by the DMA controller. When DMA is properly setup, it would read data from the pre-determined memory system and write them to the memory mapped PSA Signature Register. Each time PSA Signature Register is written a signature is generated.

CPU itself can also perform data transfer by reading from the memory system and perform write operation to PSA Signature Register if CPU has enough throughput to handle data patterns transfer.

After system reset and when AUTO mode is enabled, CRC Controller automatically generates a DMA request to request the pre-determined CRC value corresponding to the first sector of memory to be checked.

In AUTO mode, when one sector of data patterns is compressed, the signature stored at the PSA Signature Register is first copied to the PSA Sector Signature Register and PSA Signature Register is then cleared out to all zeros. An automatic signature verification is then performed by comparing the signature stored at the PSA Sector Signature Register to the CRC Value Register. After the comparison the CRC Controller can generate a DMA request. Upon receiving the DMA request the DMA controller will update the CRC Value Register by transferring the next pre-determined signature value associated with the next sector of memory system. If the signature verification fails then CRC Controller can generate a CRC fail interrupt.

In Full-CPU mode, no DMA request and interrupt are generated at all. The number of data patterns to be compressed is determined by CPU itself. Full-CPU mode is useful when DMA controller is not available to perform background data patterns transfer. The OS can periodically generate a software interrupt to CPU and use CPU to accomplish data transfer and signature verification.

CRC Controller supports doubleword, word, half word and byte access to the PSA Signature Register. During a non-doubleword write access, all unwritten byte lanes are padded with zero's before compression. Note that comparison between PSA Sector Signature Register and CRC Value Register is always in 64 bit because a compressed value is always expressed in 64 bit.

There is a software reset per channel for PSA Signature Register. When set, the PSA Signature Register is reset to all zeros.

PSA Signature Register is reset to zero under the following conditions:

- System reset
- PSA Software reset
- One sector of data patterns are compressed

#### **20.6.4.4 PSA Sector Signature Register**

After one sector of data is compressed, the final resulting signature calculated by PSA Signature Register is transferred to the PSA Sector Signature Register. PSA Signature Register is a read only register. During Semi-CPU mode, the host CPU should read from the PSA Sector Signature Register instead of reading from PSA Signature Register for signature verification to avoid data coherency issue. The PSA Signature Register can be updated with new signature before the host CPU is able to retrieve it.

In Semi-CPU mode, no DMA request is generated. When one sector of data patterns is compressed, CRC controller first generates a compression complete interrupt. Responding to the interrupt, CPU will in the ISR read the PSA Sector Signature Register and compare it to the known good signature or write the signature value to another memory location to build a signature file. In Semi-CPU mode, CPU must perform the signature verification in a manner to prevent any overrun condition. The overrun condition occurs when the compression complete interrupt is generated after one sector of data patterns is compressed and CPU has not read from the PSA Sector Signature Register to perform necessary signature verification before PSA Sector Signature Register is overridden with a new value. An overrun interrupt can be enable to generate when overrun condition occurs. During Semi-CPU mode, the host CPU should read from the PSA Sector Signature Register instead of reading from PSA Signature Register for signature verification to avoid data coherency issue. The PSA Signature Register can be updated with new signature before the host CPU is able to retrieve it.

#### 20.6.4.5 CRC Value Register

Associated with each channel there is a CRC Value Register. The CRC Value Register stores the pre-determined CRC value. After one sector of data patterns is compressed by PSA Signature Register, CRC Controller can automatically compare the resulting signature stored at the PSA Sector Signature Register with the pre-determined value stored at the CRC Value Register if AUTO mode is enabled. If the signature verification fails, CRC Controller can be enabled to generate an CRC fail interrupt. When the channel is set up for Semi-CPU mode, CRC controller first generates a compression complete interrupt to CPU. Upon servicing the interrupt, CPU will then read the PSA Sector Signature Register and then read the corresponding CRC value stored at another location and compare them. CPU should not read from the CRC Value Register during Semi-CPU or Full-CPU mode because the CRC Value Register is not updated during these two modes.

In AUTO mode, for first sector's signature, DMA request is generated when mode is programmed to AUTO. For subsequent sectors, DMA request is generated after each sector is compressed. Responding to the DMA request, DMA controller reloads the CRC Value Register for the next sector of memory system to be checked.

When CRC Value Register is updated with a new CRC value, an internal flag is set to indicate that CRC Value Register contains the most current value. This flag is cleared when CRC comparison is performed. Each time at the end of the final data pattern compression of a sector, CRC Controller first checks to see if the corresponding CRC Value Register has the most current CRC value stored in it by polling the flag. If the flag is set then the CRC comparison can be performed. If the flag is not set then it means the CRC Value Register contains stale information. A CRC underrun interrupt is generated. When an underrun condition is detected, signature verification is not performed.

CRC Controller supports doubleword, word, half word and byte access to the CRC Value Register. As noted before comparison between PSA Sector Signature Register and CRC Value Register during AUTO mode is carried out in 64 bit.

#### 20.6.4.6 Raw Data Register

The raw or un-compressed data written to the PSA Signature Register is also saved in the Raw Data Register. This register is read only.

#### 20.6.4.7 Example DMA Controller Setup

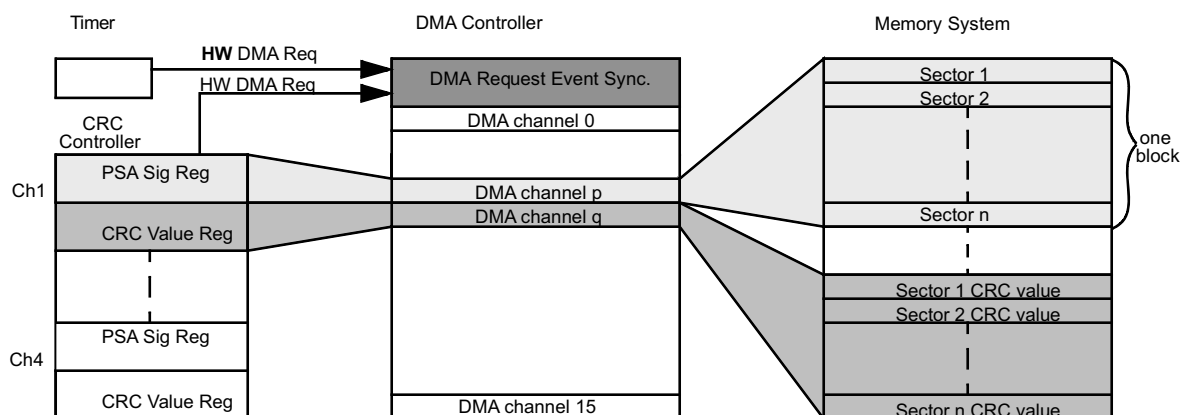
DMA controller needs to be setup properly in either either AUTO or Semi-CPU mode as DMA controller is used to transfer data patterns. Hardware or a combination of hardware and software DMA triggering are supported.

##### 20.6.4.7.1 AUTO Mode Using Hardware Timer Trigger

There are two DMA channels associated with each CRC channel when in AUTO mode. One DMA channel is setup to transfer data patterns from the source memory to the PSA Signature Register. The second DMA channel is setup to transfer the pre-determined signature to the CRC Value Register. The trigger source for the first DMA channel can be either by hardware or by software. As illustrated in [Figure 20-129](#) a timer can be used to trigger a DMA request to initiate transfer from the source memory system to PSA Signature Register. In AUTO mode, CRC Controller also generates DMA request after one sector of data patterns is compressed to initiate transfer of the next CRC value corresponding to the next sector of memory. Thus a new CRC value is always updated in the CRC Value Register by DMA synchronized to each sector of memory.

A block of memory system is usually divided into many sectors. All sectors are the same size. The sector size is programmed in the CRC\_PCOUNT\_REGx and the number of sectors in one block is programmed in the CRC\_SCOUNT\_REGx of the respective channel. CRC\_PCOUNT\_REGx multiplies CRC\_SCOUNT\_REGx and multiplies transfer size of each data pattern should give the total block size in number of bytes.

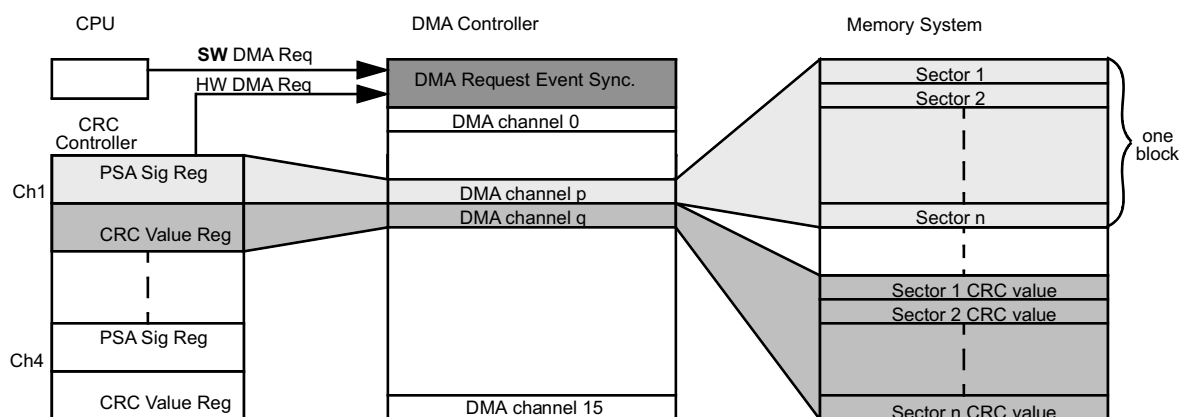
The total size of the memory system to be examined is also programmed in the respective transfer count register inside DMA module. The DMA transfer count register is divided into two parts. They are element count and frame count. Note that an HW DMA request can be programmed to trigger either one frame or one entire block transfer. In [Figure 20-129](#), an HW DMA request from a timer is used as a trigger source to initiate DMA transfer. If all two CRC channels are active in AUTO mode then a total of two DMA requests would be generated by CRC Controller.



**Figure 20-129. AUTO Mode Using Hardware Timer Trigger**

#### 20.6.4.7.2 AUTO Mode Using Software Trigger

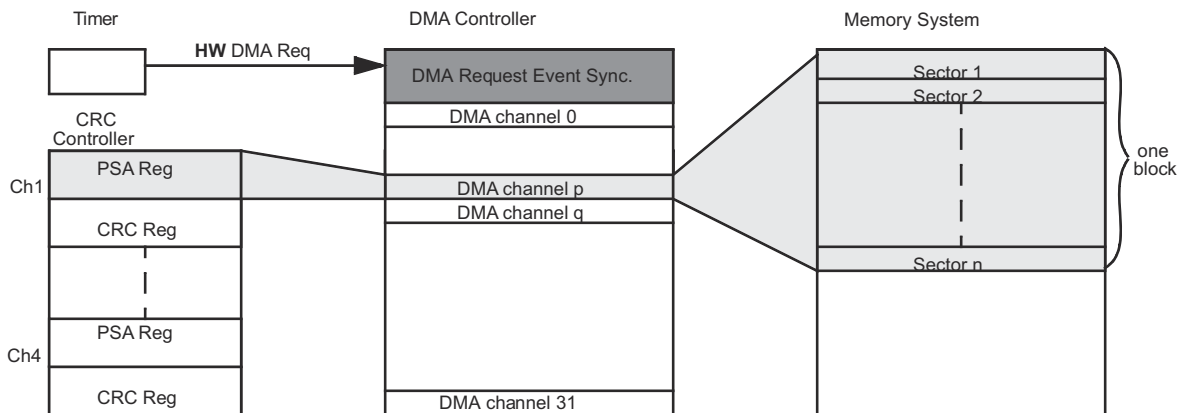
The data patterns transfer can also be initiated by software. CPU can generate a software DMA request to activate the DMA channel to transfer data patterns from source memory system to the PSA Signature Register. To generate a software DMA request CPU needs to set the corresponding DMA channel in the DMA software trigger register. Note that just one software DMA request from CPU is enough to complete the entire data patterns transfer for all sectors. See [Figure 20-130](#) for an illustration.



**Figure 20-130. AUTO Mode With Software CPU Trigger**

#### 20.6.4.7.3 Semi-CPU Mode Using Hardware Timer Trigger

During semi-CPU mode, no DMA request is generated by CRC controller. Therefore, no DMA channel is allocated to update CRC Value Register. CPU should not read from CRC Value Register in semi-CPU mode as it contains stale value. Note that no signature verification is performed at all during this mode. Similar to AUTO mode, either by hardware or by software DMA request can be used as a trigger for data patterns transfer. [Figure 20-131](#) illustrates the DMA setup using semi-CPU mode with hardware timer trigger.



**Figure 20-131. Semi-CPU Mode With Hardware Timer Trigger**

**Table 20-151. CRC Modes in Which DMA Request and Counter Logic are Active or Inactive**

Mode	DMA Request	Pattern Counter	Sector Counter	Timeout Counter
AUTO	Active	Active	Active	Active
Semi-CPU	Inactive	Active	Active	Active
Full-CPU	Inactive	Inactive	Inactive	Inactive

#### 20.6.4.8 Pattern Count Register

There is a 20-bit data pattern counter for every CRC channel. The data pattern counter is a down counter and can be pre-loaded with a programmable value stored in the Pattern Count Register. When the data pattern counter reaches zero, a compression complete interrupt is generated in Semi-CPU mode and an automatic signature verification is performed in AUTO mode. In AUTO only, DMA request is generated to trigger the DMA controller to update the CRC Value Register.

#### Note

The data pattern count should be divisible by the total transfer count as programmed in DMA controller. The total transfer count is the product of element count and frame count.

#### 20.6.4.9 Sector Count Register/Current Sector Register

Each channel contains a 16 bit sector counter. The sector count register stores the number of sectors. Sector counter is a free running counter and is incremented by one each time when one sector of data patterns is compressed. When the signature verification fails, the current value stored in the sector counter is saved into current sector register. If signature verification fails, CPU can read from the current sector register to identify the sector which causes the CRC mismatch. To aid and facilitate the CPU in determining the cause of a CRC failure, it is advisable to use the following equation during CRC and DMA setup:

$$\text{CRC Pattern Count} \times \text{CRC Sector Count} = \text{DMA Element Count} \times \text{DMA Frame Count}$$



The current sector register is frozen from being updated until both the current sector register is read and CRC fail status bit is cleared by CPU. If CPU does not respond to the CRC failure in a timely manner before another sector produces a signature verification failure, the current sector register is not updated with the new sector number. An overrun interrupt is generate instead. If current sector register is already frozen with an erroneous sector and emulation is entered with SUSPEND signal goes to high then the register still remains frozen even it is read.

In Semi-CPU mode, the current sector register is used to indicate the sector for which the compression complete has last happened.

The current sector register is reset when the PSA software reset is enabled.

### Note

Both data pattern count and sector count registers must be greater than or equal to one for the counters to count. After reset, pattern count and sector count registers default to zero and the associated counters are inactive.

#### 20.6.4.10 Interrupt

The CRC controller generates several types of interrupts per channel. Associated with each interrupt, there is an interrupt enable bit. No interrupt is generated in Full-CPU mode.

- Compression complete interrupt
- CRC fail interrupt
- Overrun interrupt
- Underrun interrupt
- Timeout interrupt

**Table 20-152. Modes in Which Interrupt Condition Can Occur**

	AUTO	Semi-CPU	Full-CPU
Compression Complete	no	yes	no
CRC Fail	yes	no	no
Overrun	yes	yes	no
Underrun	yes	no	no
Timeout	yes	yes	no

##### 20.6.4.10.1 Compression Complete Interrupt

Compression complete interrupt is generated in Semi-CPU mode only. When the data pattern counter reaches zero, the compression complete flag is set and the interrupt is generated.

##### 20.6.4.10.2 CRC Fail Interrupt

CRC fail interrupt is generated in AUTO mode only. When the signature verification fails, the CRC fail flag is set,. CPU should take action to address the fail condition and clear the CRC fail flag after it resolves the CRC mismatch.

##### 20.6.4.10.3 Overrun Interrupt

Overrun interrupt is generated in either AUTO or Semi-CPU mode. During AUTO mode, if a CRC fail is detected then the current sector number is recorded in the current sector register. If CRC fail status bit is not cleared and current sector register is not read by the host CPU before another CRC fail is detected for another sector then an overrun interrupt is generated. During Semi-CPU mode, when the data pattern counter finishes counting, it generates a compression complete interrupt. At the same time the signature is copied into the PSA Sector Signature Register. If the host CPU does not read the signature from PSA Sector Signature Register before it is updated again with a new signature value then an overrun interrupt is generated.



#### 20.6.4.10.4 Underrun Interrupt

Underrun interrupt only occurs in AUTO mode. The interrupt is generated when the CRC Value Register is not updated with the corresponding signature when the data pattern counter finishes counting. During AUTO mode, CRC Controller generates DMA request to update CRC Value Register in synchronization to the corresponding sector of the memory. Signature verification is also performed if underrun condition is detected. And CRC fail interrupt is generated at the same time as the underrun interrupt.

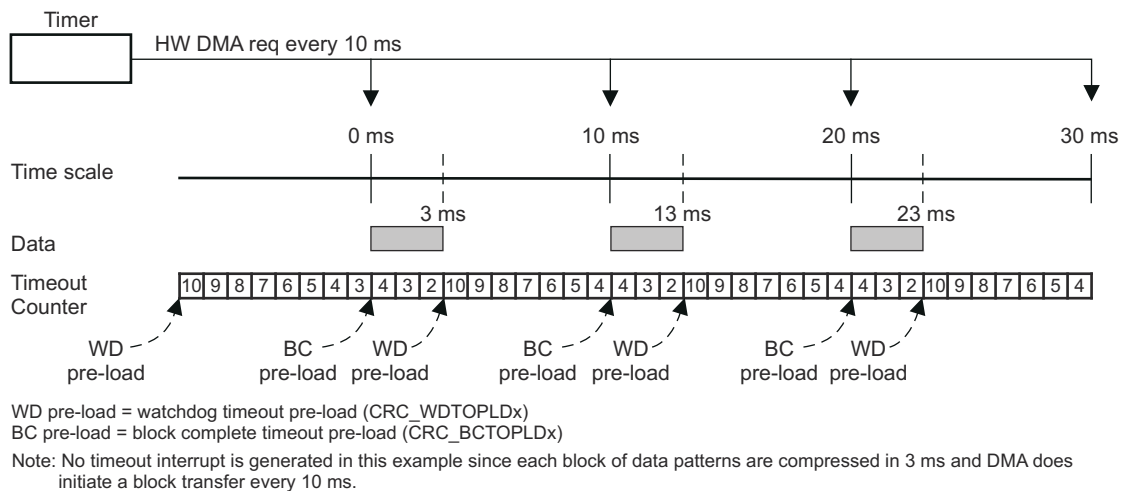
#### 20.6.4.10.5 Timeout Interrupt

To ensure that the memory system is examined within a pre-defined time frame and no loss of incoming data there is a 24 bit timeout counter per CRC channel. The 24 bit timeout down counter can be pre-loaded with two different pre-load values, watchdog timeout pre-load value (CRC\_WDTPLDx) and block complete timeout pre-load value (CRC\_BCTOPLDx). The timeout counter is clocked by a prescaler clock which is permanently running at division 64 of HCLK clock.

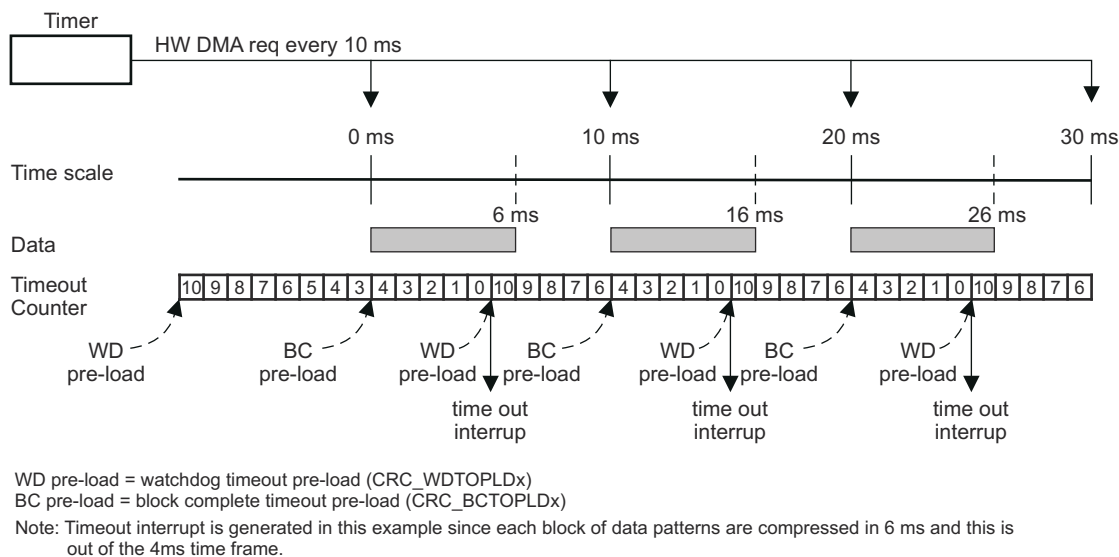
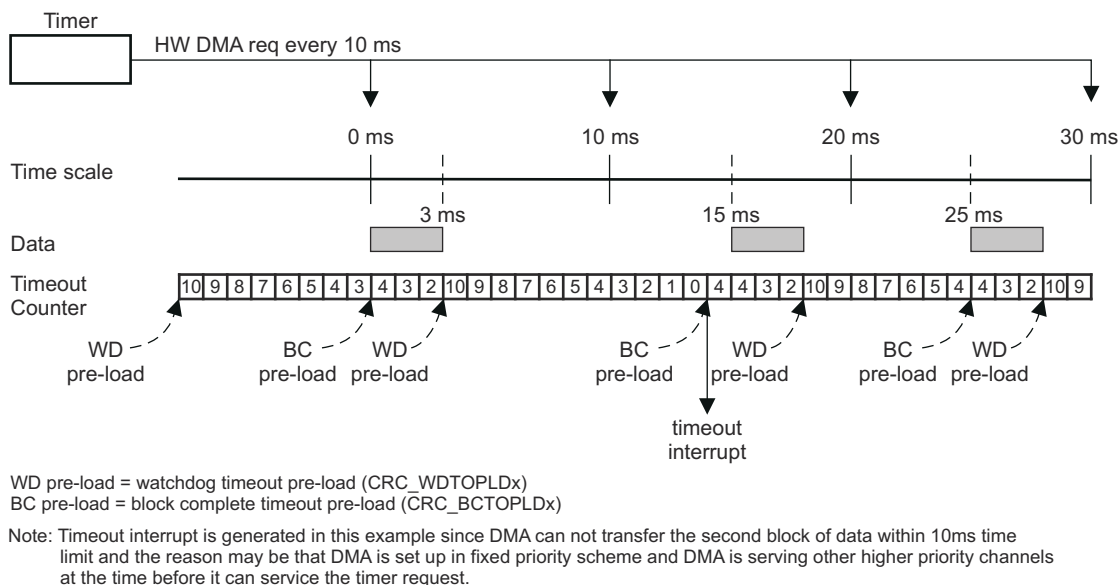
First pattern of data must be transferred by the DMA before the timeout counter expires, Watchdog timeout pre-load register (CRC\_WDTPLDx) is used as timeout counter. Block complete timeout pre-load register (CRC\_BCTOPLDx) is used to check if one complete block of data patterns are compressed within a specific time frame. The timeout counter is first pre-loaded with CRC\_WDTPLDx after either AUTO or Semi-CPU mode is selected and starts to down count. If the timeout counter expires before DMA transfers any data pattern to PSA Signature Register then a timeout interrupt is generated. An incoming data pattern before the timeout counter expires will automatically pre-load the timeout counter with CRC\_BCTOPLDx the block complete timeout pre-load value.

Block complete timeout pre-load value is used to check if one block of data patterns are compressed within a given time limit. If the timeout counter pre-loaded with CRC\_BCTOPLDx value expires before one block of data patterns are compressed a timeout interrupt is generated. When one block (pattern count x sector count) of data patterns are compressed before the counter has expired, the counter is pre-loaded with CRC\_WDTPLDx value again. If the timeout counter is pre-loaded with zero then the counter is disable and no timeout interrupt is generated.

In Figure 20-132, a timer generates DMA request every 10ms to trigger one block (pattern count x sector count) transfer. Since we want to make sure that DMA does start to transfer a block every 10 ms we would set the first pre-load value to 10ms in CRC\_WDTPLDx. We also want to make sure that one block of data patterns are compressed within 4ms. With such a requirement, we would set the second pre-load value to 4ms in CRC\_BCTOPLDx register.



**Figure 20-132. Timeout Example 1**


**Figure 20-133. Timeout Example 2**

**Figure 20-134. Timeout Example 3**

#### 20.6.4.10.6 Interrupt Offset Register

CRC Controller only generates one interrupt request to interrupt manager. A interrupt offset register is provided to indicate the source of the pending interrupt with highest priority. [Table 20-153](#) shows the offset interrupt vector address of each interrupt condition in an ascending order of priority.

**Table 20-153. Interrupt Offset Mapping**

Offset Value	Interrupt Condition
0	Phantom
1h	Ch1 CRC Fail
2h	Ch2 CRC Fail
3h-8h	Reserved
9h	Ch1 Compression Complete
Ah	Ch2 Compression Complete
Bh-10h	Reserved
11h	Ch1 Overrun
12h	Ch2 Overrun
13h-18h	Reserved
19h	Ch1 Underrun
1Ah	Ch2 Underrun
1Bh-20h	Reserved
21h	Ch1 Timeout
22h	Ch2 Timeout
23h-24h	Reserved

#### 20.6.4.10.7 Error Handling

When an interrupt is generated, host CPU should take appropriate actions to identify the source of error and restart the respective channel in DMA and CRC module. To restart a CRC channel, the user should perform the following steps in the ISR:

1. Write to software reset bit in CRC\_CTRL register to reset the respective PSA Signature Register.
2. Reset the CHx\_MODE bits to 00 in CRC\_CTRL register as Data capture mode.
3. Set the CHx\_MODE bits in CRC\_CTRL register to desired new mode again.
4. Release software reset.

The host CPU should use byte write to restart each individual channel.

#### 20.6.4.11 Power Down Mode

CRC module can be put into power down mode when the power down control bit PWDN is set. The module wakes up when the PWDN bit is cleared.

#### 20.6.4.12 Emulation

A read access from a register in functional mode can sometimes trigger a certain internal event to follow. For example, reading an interrupt offset register triggers an event to clear the corresponding interrupt status flag. During emulation when SUSPEND signal is high, a read access from any register should only return the register contents to the bus and should not trigger or mask any event as it would have in functional mode. This is to prevent debugger from reading the interrupt offset register during refreshing screen and cause the corresponding interrupt status flag to get cleared. Timeout counters are stopped to generate timeout interrupts in emulation mode. No Peripheral Controller bus error should be generated if reading from the unimplemented locations.

#### 20.6.4.13 Peripheral Bus Interface

CRC is a Peripheral target module. The register interface is similar to other peripheral modules. CRC supports following features:

- Different sizes of burst operation.
- Aligned and unaligned accesses.
- Abort is generated for any illegal address accesses.

#### 20.6.5 Example

This section illustrates several of the ways in which the CRC Controller can be utilized to perform CRC.

##### 20.6.5.1 Example: Auto Mode Using Time Based Event Triggering

A large memory area with 2Mbyte (256k doubleword) is to be checked in the background of CPU. CRC is to be performed every 1K byte (128 doubleword). Therefore there should be 2048 pre-recorded CRC values. For illustration purpose, we map channel 1 CRC Value Register to DMA channel 1 and channel 1 PSA Signature Register to DMA channel 2. Assume all DMA transfers are carried out in 64-bit transfer size.

##### 20.6.5.1.1 DMA Setup

- Set up DMA channel 1 with the starting address from which the pre-determined CRC values are stored. Set up the destination address to the memory mapped channel 1 CRC Value Register. Put the source address at post increment addressing mode and put the destination address at constant addressing mode. Use **hardware** DMA request for channel 1 to trigger a **frame** transfer.
- Set up DMA channel 2 with the source address from which the contents of memory to be verified. Set up the destination address to the memory mapped channel 1 PSA Signature Register. Program the element transfer count to 128 and the frame transfer count to 2048. Put the source address at post increment addressing mode and put the destination address at constant address mode. Use **hardware** DMA request for channel 2 to trigger an entire **block** transfer.

##### 20.6.5.1.2 Timer Setup

The timer can be any general purpose timer which is capable of generating a time-based DMA request.

- Set up timer to generate DMA request associated with DMA channel 2. For example, an OS can set up the timer to generate a DMA request every 10ms.

### 20.6.5.1.3 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 2048.
- Enable AUTO mode and all interrupts.

After AUTO mode is selected, CRC Controller automatically generates a DMA request on channel 1. Around the same time the timer module also generates a DMA request on DMA channel 2. When the first incoming data pattern arrives at the PSA Signature Register, the CRC Controller will compress it. After some time, the DMA controller would update the CRC Value Register with a pre-determined value matching the calculated signature for the first sector of 128 64 bit data patterns. After one sector of data patterns are compressed, the CRC Controller generate a CRC fail interrupt if signature stored at the PSA Sector Signature Register does not match the CRC Value Register. CRC Controller generates a DMA request on DMA channel 1 when one sector of data patterns are compressed. This routine will continue until the entire 2Mbyte are consumed. If the timeout counter reached zero before the entire 2Mbytes are compressed a timeout interrupt is generated. After 2MBytes are transferred, the DMA can generate an interrupt to CPU. The entire operation will continue again when DMA responds to the DMA request from both the timer and CRC Controller. The CRC is performed totally without any CPU intervention.

### 20.6.5.2 Example: Auto Mode Without Using Time Based Triggering

A small but highly secured memory area with 1kbytes is to be checked in the background of CPU. CRC is to be performed every 1Kbytes. Therefore there is only one pre-recorded CRC value. For illustration purpose, we map channel 1 CRC Value Register to DMA channel 1 and channel 1 PSA Signature Register to DMA channel 2. Assume all transfers carried out by DMA are in 64 bit transfer size.

#### 20.6.5.2.1 DMA Setup

- Set up DMA channel 1 with the source address from which the pre-determined CRC value is stored. Set up the destination address to the memory mapped channel 1 CRC Value Register. Put the source address at constant addressing mode and put the destination address at constant addressing mode. Use **hardware** DMA request for channel 1.
- Set up DMA channel 2 with the source address from which the memory area to be verified. Set up the destination address to the memory mapped channel 1 PSA Signature Register. Program the element transfer count to 128 and the frame transfer count to 1. Put the source address at post increment addressing mode and put the destination address at constant address mode. Generate a **software** DMA request on channel 2 after CRC has completed its setup. Enable autoinitiation for DMA channel 2.

#### 20.6.5.2.2 CRC Setup

- Program the pattern count to 128.
- Program the sector count to 1.
- Leaving the timeout count register with the reset value of zero means no timeout interrupt is generated.
- Enable AUTO mode and all interrupts.

After AUTO mode is selected, the CRC Controller automatically generates a DMA request on channel 1. At the same time the CPU generates a **software** DMA request on DMA channel 2. When the first incoming data pattern arrives at the PSA Signature Register, the CRC Controller will compress it. After some time, the DMA controller would update the CRC Value Register with a pre-determined value matching the calculated signature for the first sector of 128 64 bit data patterns. After one sector of data patterns are compressed, the CRC Controller generates a CRC fail interrupt if signature stored at the PSA Sector Signature Register does not match the CRC Value Register. CRC Controller generates a DMA request on DMA channel 1 again after one sector is compressed. After 1kbytes are transferred, the DMA can generate an interrupt to CPU. Responding to the DMA interrupt CPU can restart the CRC routine by generating a software DMA request onto channel 2 again.

### 20.6.5.3 Example: Semi-CPU Mode

If DMA controller is available in a system, the CRC module can also operate in semi-CPU mode. This means that CPU can still make use of the DMA to perform data patterns transfer to CRC controller in the background. The difference between semi-CPU mode and AUTO mode is that CRC controller does not automatically perform

the signature verification. CRC controllers generates a compression complete interrupt to CPU when the one sector of data patterns are compressed. CPU needs to perform the signature verification itself.

A memory area with 2Mbyte is to be verified with the help of the CPU. CRC operation is to be performed every 1K byte. Since there are 2Mbyte (256k doublewords) of memory to be check and we want to perform a CRC every 1Kbyte (128 doublewords) and therefore there should be 2048 pre-recorded CRC values. In Semi-CPU mode, the CRC Value Register is not updated and contains indeterminate data.

#### **20.6.5.3.1 DMA Setup**

Set up DMA channel 1 with the source address from which the memory area to be verified are mapped. Set up the destination address to the memory mapped channel 1 PSA Signature Register. Put the starting address at post increment addressing mode and put the destination address at constant address mode. Use hardware DMA request to trigger an entire block transfer for channel 1. Disable autoinitiation for DMA channel 1.

#### **20.6.5.3.2 Timer Setup**

The timer can be any general purpose timer which is capable of generating a time based DMA request.

Set up timer to generate DMA request associated with DMA channel 1. For example, an OS can set up the timer to generate a DMA request every 10ms.

#### **20.6.5.3.3 CRC Setup**

- Program the pattern count to 128.
- Program the sector count to 2048.
- Enable Semi-CPU mode and enable all interrupts.

The timer module first generates a DMA request on DMA channel 1 when it is enabled. When the first incoming data pattern arrives at the PSA Signature Register, the CRC controller will compress it. After one sector of data patterns are compressed, the CRC controller generate a compression complete interrupt. Upon responding to the interrupt the CPU would read from the PSA Sector Signature Register. It is up to the CPU on how to deal with the PSA value just read. It can compare it to a known signature value or it can write it to another memory location to build a signature file or even transfer the signature out of the device via SCI or SPI. This routine will continue until the entire 2Mbyte are consumed. The latency of the interrupt response from CPU can cause overrun condition. If CPU does not read from PSA Sector Signature Register before the PSA value is overridden with the signature of the next sector of memory, an overrun interrupt will be generated by CRC controller.

#### **20.6.5.4 Example: Full-CPU Mode**

In a system without the availability of DMA controller, the CRC routine can be operated by CPU provided the CPU has enough throughput. CPU needs to read from the memory area from which CRC is to be performed.

A memory area with 2Mbyte is to be checked with the help of the CPU. CRC verification is to be performed every 1K byte. In CPU mode, the CRC Value Register is not updated and contains indeterminate data.

##### **20.6.5.4.1 CRC Setup**

- All control registers can be left in their reset state. Only enable Full-CPU mode.

CPU itself reads from the memory and write the data to the PSA Signature Register inside CRC Controller. When the first incoming data pattern arrives at the PSA Signature Register, the CRC Controller will compress it. After **2MBytes** data patterns are compressed, CPU can read from the PSA Signature Register. It is up to the CPU on how to deal with the PSA signature value just read. It can compare it to a known signature value stored at another memory location.

## 20.6.6 MSS\_MCRC Registers

Table 20-154 lists the PCR\_generated\_memory\_map registers. All register offset addresses not listed in Table 20-154 should be considered as reserved locations and the register contents should not be modified.

**Table 20-154. MSS\_MCRC Registers**

Offset	Acronym	Register Name	Section
0h	CRC_CTRL0	CRC Global Control Register 0	<a href="#">Section 20.6.6.1</a>
8h	CRC_CTRL1	CRC Global Control Register 1	<a href="#">Section 20.6.6.2</a>
10h	CRC_CTRL2	CRC Global Control Register 2	<a href="#">Section 20.6.6.3</a>
18h	CRC_INTS	CRC Interrupt Enable Set Register	<a href="#">Section 20.6.6.4</a>
20h	CRC_INTR	CRC Interrupt Enable Reset Register	<a href="#">Section 20.6.6.5</a>
28h	CRC_STATUS_REG	CRC Interrupt Status Register-	<a href="#">Section 20.6.6.6</a>
30h	CRC_INT_OFFSET_REG	CRC Interrupt Offset	<a href="#">Section 20.6.6.7</a>
38h	CRC_BUSY	CRC Busy Register during AUTO mode	<a href="#">Section 20.6.6.8</a>
40h	CRC_PCOUNT_REG1	CRC Pattern Counter Pre-load Register1	<a href="#">Section 20.6.6.9</a>
44h	CRC_SCOUNT_REG1	CRC Sector Counter Pre-load Register1	<a href="#">Section 20.6.6.10</a>
48h	CRC_CURSEC_REG1	CRC Current Sector Register 1	<a href="#">Section 20.6.6.11</a>
4Ch	CRC_WDTPD1	CRC channel 1 Watchdog Timeout Preload Register A	<a href="#">Section 20.6.6.12</a>
50h	CRC_BCTOPD1	CRC channel 1 Block Complete Timeout Preload Register B	<a href="#">Section 20.6.6.13</a>
60h	PSA_SIGREGL1	Channel 1 PSA signature low register	<a href="#">Section 20.6.6.14</a>
64h	PSA_SIGREGH1	Channel 1 PSA signature high register	<a href="#">Section 20.6.6.15</a>
68h	CRC_REGL1	Channel 1 CRC value low register	<a href="#">Section 20.6.6.16</a>
6Ch	CRC_REGH1	Channel 1 CRC value high register	<a href="#">Section 20.6.6.17</a>
70h	PSA_SECSIGREGL1	Channel 1 PSA sector signature low register	<a href="#">Section 20.6.6.18</a>
74h	PSA_SECSIGREGH1	Channel 1 PSA sector signature high register	<a href="#">Section 20.6.6.19</a>
78h	RAW_DATAAREGL1	Channel 1 Raw Data Low Register	<a href="#">Section 20.6.6.20</a>
7Ch	RAW_DATAAREGH1	Channel 1 Raw Data High Register	<a href="#">Section 20.6.6.21</a>
80h	CRC_PCOUNT_REG2	CRC Pattern Counter Pre-load Register2	<a href="#">Section 20.6.6.22</a>
84h	CRC_SCOUNT_REG2	CRC Sector Counter Pre-load Register2	<a href="#">Section 20.6.6.23</a>
88h	CRC_CURSEC_REG2	CRC Current Sector Register 2	<a href="#">Section 20.6.6.24</a>
8Ch	CRC_WDTPD2	CRC channel 2 Watchdog Timeout Preload Register	<a href="#">Section 20.6.6.25</a>
90h	CRC_BCTOPD2	CRC channel 2 Block Complete Timeout Preload Register	<a href="#">Section 20.6.6.26</a>
A0h	PSA_SIGREGL2	Channel 2 PSA signature low register	<a href="#">Section 20.6.6.27</a>
A4h	PSA_SIGREGH2	Channel 2 PSA signature high register	<a href="#">Section 20.6.6.28</a>
A8h	CRC_REGL2	Channel 2 CRC value low register	<a href="#">Section 20.6.6.29</a>
ACh	CRC_REGH2	Channel 2 CRC value high register	<a href="#">Section 20.6.6.30</a>
B0h	PSA_SECSIGREGL2	Channel 2 PSA sector signature low register	<a href="#">Section 20.6.6.31</a>
B4h	PSA_SECSIGREGH2	Channel 2 PSA sector signature high register	<a href="#">Section 20.6.6.32</a>
B8h	RAW_DATAAREGL2	Channel 2 Raw Data Low Register	<a href="#">Section 20.6.6.33</a>
BCh	RAW_DATAAREGH2	Channel 2 Raw Data High register	<a href="#">Section 20.6.6.34</a>
C0h	CRC_PCOUNT_REG3	CRC Pattern Counter Pre-load Register3	<a href="#">Section 20.6.6.35</a>
C4h	CRC_SCOUNT_REG3	CRC Sector Counter Pre-load Register3	<a href="#">Section 20.6.6.36</a>
C8h	CRC_CURSEC_REG3	CRC Current Sector Register 3	<a href="#">Section 20.6.6.37</a>
CCh	CRC_WDTPD3	CRC channel 3 Watchdog Timeout Preload Register	<a href="#">Section 20.6.6.38</a>
D0h	CRC_BCTOPD3	CRC channel 3 Block Complete Timeout Preload Register	<a href="#">Section 20.6.6.39</a>



**Table 20-154. MSS\_MCRC Registers (continued)**

Offset	Acronym	Register Name	Section
E0h	PSA_SIGREGL3	Channel 3 PSA signature low register	<a href="#">Section 20.6.6.40</a>
E4h	PSA_SIGREGH3	Channel 3 PSA signature high register	<a href="#">Section 20.6.6.41</a>
E8h	CRC_REGL3	Channel 3 CRC value low register	<a href="#">Section 20.6.6.42</a>
ECh	CRC_REGH3	Channel 3 CRC value high register	<a href="#">Section 20.6.6.43</a>
F0h	PSA_SECSIGREGL3	Channel 3 PSA sector sig-nature low register	<a href="#">Section 20.6.6.44</a>
F4h	PSA_SECSIGREGH3	Channel 3 PSA sector sig-nature high register	<a href="#">Section 20.6.6.45</a>
F8h	RAW_DATAAREGL3	Channel 3 Raw Data Low Register	<a href="#">Section 20.6.6.46</a>
FCh	RAW_DATAAREGH3	Channel 3 Raw Data High register	<a href="#">Section 20.6.6.47</a>
100h	CRC_PCOUNT_REG4	CRC Pattern Counter Pre-load Register4	<a href="#">Section 20.6.6.48</a>
104h	CRC_SCOUNT_REG4	CRC Sector Counter Pre-load Register4	<a href="#">Section 20.6.6.49</a>
108h	CRC_CURSEC_REG4	CRC Current Sector Regis-ter 4	<a href="#">Section 20.6.6.50</a>
10Ch	CRC_WDTPLD4	CRC channel 4 Watchdog Timeout Preload Register	<a href="#">Section 20.6.6.51</a>
110h	CRC_BCTOPLD4	CRC channel 4 Block Com-plete Timeout Preload Reg-ister	<a href="#">Section 20.6.6.52</a>
120h	PSA_SIGREGL4	Channel 4 PSA signature low register	<a href="#">Section 20.6.6.53</a>
124h	PSA_SIGREGH4	Channel 4 PSA signature high register	<a href="#">Section 20.6.6.54</a>
128h	CRC_REGL4	Channel 4 CRC value low register	<a href="#">Section 20.6.6.55</a>
12Ch	CRC_REGH4	Channel 4 CRC value high register	<a href="#">Section 20.6.6.56</a>
130h	PSA_SECSIGREGL4	Channel 4 PSA sector sig-nature low register	<a href="#">Section 20.6.6.57</a>
134h	PSA_SECSIGREGH4	Channel 4 PSA sector sig-nature high register	<a href="#">Section 20.6.6.58</a>
138h	RAW_DATAAREGL4	Channel 4 Raw Data Low Register	<a href="#">Section 20.6.6.59</a>
13Ch	RAW_DATAAREGH4	Channel 4 Raw Data High register	<a href="#">Section 20.6.6.60</a>
140h	MCRC_BUS_SEL	Data bus tracing selection	<a href="#">Section 20.6.6.61</a>
144h	MCRC_RESERVED	RESERVED	<a href="#">Section 20.6.6.62</a>



### 20.6.6.1 CRC\_CTRL0 Register (Offset = 0h) [reset = 0h]

CRC\_CTRL0 is shown in [Figure 20-135](#) and described in [Table 20-155](#).

Return to the [Table 20-154](#).

Contains sw reset control bit to reset PSA

**Figure 20-135. CRC\_CTRL0 Register**

31	30	29	28	27	26	25	24
NU12	NU11	NU10	NU9	NU8	NU7		
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
NU6	NU5	NU4	NU3	NU2	NU1		
R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
CH2_CRC_SEL 2	CH2_BYTE_S WAP	CH2_BIT_SWA P	CH2_CRC_SEL	CH2_DW_SEL	CH2_PSA_SW REST		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h
7	6	5	4	3	2	1	0
CH1_CRC_SEL 2	CH1_BYTE_S WAP	CH1_BIT_SWA P	CH1_CRC_SEL	CH1_DW_SEL	CH1_PSA_SW REST		
R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h	R/W-0h

**Table 20-155. CRC\_CTRL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	NU12	R	0h	Reserved
30	NU11	R	0h	Reserved
29	NU10	R	0h	Reserved
28-27	NU9	R	0h	Reserved
26-25	NU8	R	0h	Reserved
24	NU7	R	0h	Reserved
23	NU6	R	0h	Reserved
22	NU5	R	0h	Reserved
21	NU4	R	0h	Reserved
20-19	NU3	R	0h	Reserved
18-17	NU2	R	0h	Reserved
16	NU1	R	0h	Reserved
15	CH2_CRC_SEL2	R/W	0h	Refer "CH2_DW_SEL" field description
14	CH2_BYTE_SWAP	R/W	0h	BYTE SWAP Enable across Data Size 0 – Byte Swap Disabled 1 – Byte Swap enabled.
13	CH2_BIT_SWAP	R/W	0h	msb/lbs SWAPPING 0 – msb (most significant bit First) 1 – lsb (least significant bit First)
12-11	CH2_CRC_SEL	R/W	0h	CRC type select. {CH1_CRC_SEL2,CH1_CRC_SEL[1:0]} 000 – CRC-64 001 - CRC-16 010 – CRC-32 100 - VDA CAN, CRC-8, Autosar 4.0 110 - CASTAGNOLI, iSCSI 111 / 011 - E2E Profile 4
10-9	CH2_DW_SEL	R/W	0h	CRC Data Size select. 000 – 64 bit Data Size 001 - 16 bit Data Size 010 – 32 Bit Data Size

**Table 20-155. CRC\_CTRL0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
8	CH2_PSA_SWREST	R/W	0h	Channel 2 PSA Software Reset. When set, the PSA Signature Register is reset to all zero. Software reset does not reset software reset bit itself. Therefore, CPU is required to clear this bit by writing a '0'. 0 = PSA Signature Register not reset 1 = PSA Signature Register reset
7	CH1_CRC_SEL2	R/W	0h	Refer "CH1_DW_SEL" field description
6	CH1_BYTE_SWAP	R/W	0h	BYTE SWAP Enable across Data Size 0 – Byte Swap Disabled 1 – Byte Swap enabled.
5	CH1_BIT_SWAP	R/W	0h	msb/lb SWAPPING 0 – msb (most significant bit First) 1 – lsb (least significant bit First)
4-3	CH1_CRC_SEL	R/W	0h	CRC type select. {CH1_CRC_SEL2,CH1_CRC_SEL[1:0]} 000 – CRC-64 001 - CRC-16 010 – CRC-32 100 - VDA CAN, SAE-J1850 CRC-8 101 - H2F, Autosar 4.0 110 - CASTAGNOLI, iSCSI 111 / 011 - E2E Profile 4
2-1	CH1_DW_SEL	R/W	0h	CRC Data Size select. 000 – 64 bit Data Size 001 - 16 bit Data Size 010 – 32 Bit Data Size
0	CH1_PSA_SWREST	R/W	0h	Channel 1 PSA Software Reset. When set, the PSA Signature Register is reset to all zero. Software reset does not reset software reset bit itself. Therefore, CPU is required to clear this bit by writing a '0'. 0 = PSA Signature Register not reset 1 = PSA Signature Register reset

### 20.6.6.2 CRC\_CTRL1 Register (Offset = 8h) [reset = 0h]

CRC\_CTRL1 is shown in [Figure 20-136](#) and described in [Table 20-156](#).

Return to the [Table 20-154](#).

Contains power down control bit

**Figure 20-136. CRC\_CTRL1 Register**

31	30	29	28	27	26	25	24
RESERVED							
R-0h							
23	22	21	20	19	18	17	16
RESERVED							
R-0h							
15	14	13	12	11	10	9	8
RESERVED							
R-0h							
7	6	5	4	3	2	1	0
RESERVED							PWDN
R-0h							R/W-0h

**Table 20-156. CRC\_CTRL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	RESERVED	R	0h	
0	PWDN	R/W	0h	Power Down. When set, MCRC moduleMCRC Module is put in power down mode. 0 = MCRC is not in power down mode 1 = MCRC is in power down mode

### 20.6.6.3 CRC\_CTRL2 Register (Offset = 10h) [reset = 0h]

CRC\_CTRL2 is shown in [Figure 20-137](#) and described in [Table 20-157](#).

Return to the [Table 20-154](#).

Contains channel mode, data trace enable control bits

**Figure 20-137. CRC\_CTRL2 Register**

31	30	29	28	27	26	25	24
RESERVED						NU14	
R-0h						R-0h	
23	22	21	20	19	18	17	16
RESERVED						NU13	
R-0h						R-0h	
15	14	13	12	11	10	9	8
RESERVED						CH2_MODE	
R-0h						R/W-0h	
7	6	5	4	3	2	1	0
RESERVED			CH1_TRACEE N	RESERVED		CH1_MODE	
R-0h			R/W-0h	R-0h		R/W-0h	

**Table 20-157. CRC\_CTRL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	RESERVED	R	0h	
25-24	NU14	R	0h	Reserved
23-18	RESERVED	R	0h	
17-16	NU13	R	0h	Reserved
15-10	RESERVED	R	0h	
9-8	CH2_MODE	R/W	0h	Channel 2 Mode: 0 0 = Data Capture mode. In this mode, the PSA Signature Register does not compress data when it is written. Any data written to PSA Signature Register is simply captured by PSA Signature Register without any compression. This mode can be used to plant seed value into the PSA register 0 1 = AUTO mode 1 0 = Semi-CPU mode 1 1 = Full-CPU mode
7-5	RESERVED	R	0h	
4	CH1_TRACEEN	R/W	0h	Channel 1 Data Trace Enable. When set, the channel is put into data trace mode. The channel snoops on the CPU VBUSM, ITCM, DTCM buses for any read transaction. Any read data on these buses is compressed by the PSA Signature Register. When suspend is on, the PSA Signature Register does not compress any read data on these buses. 0 = Data Trace disable 1 = Data Trace enable
3-2	RESERVED	R	0h	
1-0	CH1_MODE	R/W	0h	Channel 1 Mode: 0 0 = Data Capture mode. In this mode, the PSA Signature Register does not compress data when it is written. Any data written to PSA Signature Register is simply captured by PSA Signature Register without any compression. This mode can be used to plant seed value into the PSA register 0 1 = AUTO mode 1 0 = Semi-CPU mode 1 1 = Full-CPU mode

#### 20.6.6.4 CRC\_INTS Register (Offset = 18h) [reset = 0h]

CRC\_INTS is shown in [Figure 20-138](#) and described in [Table 20-158](#).

Return to the [Table 20-154](#).

Write one to a bit to enable a interrupt

**Figure 20-138. CRC\_INTS Register**

31	30	29	28	27	26	25	24
RESERVED			NU22	NU21	NU20	NU19	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			NU18	NU17	NU16	NU15	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			CH2_TIMEOUT ENS	CH2_UNDERE NS	CH2_OVEREN S	CH2_CRCFAIL ENS	CH2_CCITENS
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			CH1_TIMEOUT ENS	CH1_UNDERE NS	CH1_OVEREN S	CH1_CRCFAIL ENS	CH1_CCITENS
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

**Table 20-158. CRC\_INTS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	NU22	R	0h	Reserved
27	NU21	R	0h	Reserved
26	NU20	R	0h	Reserved
25	NU19	R	0h	Reserved
24-21	RESERVED	R	0h	
20	NU18	R	0h	Reserved
19	NU17	R	0h	Reserved
18	NU16	R	0h	Reserved
17	NU15	R	0h	Reserved
16-13	RESERVED	R	0h	
12	CH2_TIMEOUTENS	R/W	0h	Channel 2 Timeout Interrupt Enable Bit. Writing a one to this bit enable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt enable
11	CH2_UNDERENS	R/W	0h	Channel 2 Underrun Interrupt Enable Bit. Writing a one to this bit enable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt enable

**Table 20-158. CRC\_INTS Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
10	CH2_OVERENS	R/W	0h	Channel 2 Overrun Interrupt Enable Bit. Writing a one to this bit enable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt enable
9	CH2_CRCFAILENS	R/W	0h	Channel 2 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable
8	CH2_CRC_FAILENS	R/W	0h	Channel 2 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable
7-5	RESERVED	R	0h	
4	CH1_TIMEOUTENS	R/W	0h	Channel 1 Timeout Interrupt Enable Bit. Writing a one to this bit enable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt enable
3	CH1_UNDERENS	R/W	0h	Channel 1 Underrun Interrupt Enable Bit. Writing a one to this bit enable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt enable
2	CH1_OVERENS	R/W	0h	Channel 1 Overrun Interrupt Enable Bit. Writing a one to this bit enable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt enable
1	CH1_CRCFAILENS	R/W	0h	Channel 1 CRC Fail Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt enable

**Table 20-158. CRC\_INTS Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	CH1_CCITENS	R/W	0h	Channel 1 Compression Complete Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Compression Complete Interrupt disable 1 = Compression Complete Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Compression Complete Interrupt enable

### 20.6.6.5 CRC\_INTR Register (Offset = 20h) [reset = 0h]

CRC\_INTR is shown in [Figure 20-139](#) and described in [Table 20-159](#).

Return to the [Table 20-154](#).

Write one to a bit to disable a interrupt

**Figure 20-139. CRC\_INTR Register**

31	30	29	28	27	26	25	24
RESERVED			NU30	NU29	NU28	NU27	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			NU26	NU25	NU24	NU23	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			CH2_TIMEOUT ENR	CH2_UNDERE NR	CH2_OVEREN R	CH2_CRCFAIL ENR	CH2_CCITENS
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			CH1_TIMEOUT ENR	CH1_UNDERE NR	CH1_OVEREN R	CH1_CRCFAIL ENR	CH1_CCITENS
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

**Table 20-159. CRC\_INTR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	NU30	R	0h	Reserved
27	NU29	R	0h	Reserved
26	NU28	R	0h	Reserved
25	NU27	R	0h	Reserved
24-21	RESERVED	R	0h	
20	NU26	R	0h	Reserved
19	NU25	R	0h	Reserved
18	NU24	R	0h	Reserved
17	NU23	R	0h	Reserved
16-13	RESERVED	R	0h	
12	CH2_TIMEOUTENR	R/W	0h	Channel 2 Timeout Interrupt Disable Bit. Writing a one to this bit disable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt disable
11	CH2_UNDERENR	R/W	0h	Channel 2 Underrun Interrupt Disable Bit. Writing a one to this bit disable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/dis-able). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt disable



**Table 20-159. CRC\_INTR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
10	CH2_OVERENR	R/W	0h	Channel 2 Overrun Interrupt Disable Bit. Writing a one to this bit disable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt disable
9	CH2_CRCFAILENR	R/W	0h	Channel 2 CRC Fail Interrupt Disable Bit. Writing a one to this bit disable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt disable
8	CH2_CCITENS	R/W	0h	Channel 2 Compression Complete Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Compression Complete Interrupt disable 1 = Compression Complete Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Compression Complete Interrupt enable
7-5	RESERVED	R	0h	
4	CH1_TIMEOUTENR	R/W	0h	Channel 1 Timeout Interrupt Disable Bit. Writing a one to this bit disable the timeout interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Timeout Interrupt disable 1 = Timeout Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Timeout Interrupt disable
3	CH1_UNDERENR	R/W	0h	Channel 1 Underrun Interrupt Disable Bit. Writing a one to this bit disable the underrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Underrun Interrupt disable 1 = Underrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Underrun Interrupt disable
2	CH1_OVERENR	R/W	0h	Channel 1 Overrun Interrupt Disable Bit. Writing a one to this bit disable the overrun interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Overrun Interrupt disable 1 = Overrun Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Overrun Interrupt disable
1	CH1_CRCFAILENR	R/W	0h	Channel 1 CRC Fail Interrupt Disable Bit. Writing a one to this bit disable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = CRC Fail Interrupt disable 1 = CRC Fail Interrupt enable User and privileged mode write: 0 = Has no effect 1 = CRC Fail Interrupt disable

**Table 20-159. CRC\_INTR Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
0	CH1_CCITENS	R	0h	Channel 1 Compression Complete Interrupt Enable Bit. Writing a one to this bit enable the CRC fail interrupt. Writing a zero has no effect. Reading from this bit gives the status (interrupt enable/disable). User and privileged mode read: 0 = Compression Complete Interrupt disable 1 = Compression Complete Interrupt enable User and privileged mode write: 0 = Has no effect 1 = Compression Complete Interrupt enable

### 20.6.6.6 CRC\_STATUS\_REG Register (Offset = 28h) [reset = 0h]

CRC\_STATUS\_REG is shown in [Figure 20-140](#) and described in [Table 20-160](#).

Return to the [Table 20-154](#).

Contains interrupt flags for different types of interrupt

**Figure 20-140. CRC\_STATUS\_REG Register**

31	30	29	28	27	26	25	24
RESERVED			NU38	NU37	NU36	NU35	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
23	22	21	20	19	18	17	16
RESERVED			NU34	NU33	NU32	NU31	RESERVED
R-0h			R-0h	R-0h	R-0h	R-0h	R-0h
15	14	13	12	11	10	9	8
RESERVED			CH2_TIMEOUT	CH2_UNDER	CH2_OVER	CH2_CRCFAIL	CH2_CCIT
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h
7	6	5	4	3	2	1	0
RESERVED			CH1_TIMEOUT	CH1_UNDER	CH1_OVER	CH1_CRCFAIL	CH1_CCIT
R-0h			R/W-0h	R/W-0h	R/W-0h	R/W-0h	R-0h

**Table 20-160. CRC\_STATUS\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-29	RESERVED	R	0h	
28	NU38	R	0h	Reserved
27	NU37	R	0h	Reserved
26	NU36	R	0h	Reserved
25	NU35	R	0h	Reserved
24-21	RESERVED	R	0h	
20	NU34	R	0h	Reserved
19	NU33	R	0h	Reserved
18	NU32	R	0h	Reserved
17	NU31	R	0h	Reserved
16-13	RESERVED	R	0h	
12	CH2_TIMEOUT	R/W	0h	Channel 2 CRC Timeout Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode. 0 = No timeout interrupt is active 1 = Timeout interrupt is active
11	CH2_UNDER	R/W	0h	Channel 2 CRC Underrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only 0 = No underrun interrupt is active 1 = Underrun interrupt is active
10	CH2_OVER	R/W	0h	Channel 2 CRC Overrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode 0 = No overrun interrupt is active 1 = Overrun interrupt is active

**Table 20-160. CRC\_STATUS\_REG Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
9	CH2_CRCFAIL	R/W	0h	Channel 2 CRC Compare Fail Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only. 0 = No CRC compare fail interrupt is active 1 = CRC compare fail interrupt is active
8	CH2_CCIT	R/W	0h	Channel 2 CRC Pattern Compression Complete Status Flag. This bit is cleared by writing a 1 to it only. Writing 0 has no effect. This bit is only set in Semi-CPU mode. 0 = No CRC pattern compression complete interrupt is active 1 = CRC pattern compression complete interrupt is active
7-5	RESERVED	R	0h	
4	CH1_TIMEOUT	R/W	0h	Channel 1 CRC Timeout Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode. 0 = No timeout interrupt is active 1 = Timeout interrupt is active
3	CH1_UNDER	R/W	0h	Channel 1 CRC Underrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only 0 = No underrun interrupt is active 1 = Underrun interrupt is active
2	CH1_OVER	R/W	0h	Channel 1 CRC Overrun Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode 0 = No overrun interrupt is active 1 = Overrun interrupt is active
1	CH1_CRCFAIL	R/W	0h	Channel 1 CRC Compare Fail Status Flag. This bit is cleared by writing a '1' to it only. Writing '0' has no effect. This bit is set in AUTO mode only. 0 = No CRC compare fail interrupt is active 1 = CRC compare fail interrupt is active
0	CH1_CCIT	R	0h	Channel 1 CRC Pattern Compression Complete Status Flag. This bit is cleared by writing a 1 to it only. Writing 0 has no effect. This bit is only set in Semi-CPU mode. 0 = No CRC pattern compression complete interrupt is active 1 = CRC pattern compression complete interrupt is active

### 20.6.6.7 CRC\_INT\_OFFSET\_REG Register (Offset = 30h) [reset = 0h]

CRC\_INT\_OFFSET\_REG is shown in [Figure 20-141](#) and described in [Table 20-161](#).

Return to the [Table 20-154](#).

Contains the interrupt offset vector address

**Figure 20-141. CRC\_INT\_OFFSET\_REG Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OFSTREG							
R-0h																								R/W-0h							

**Table 20-161. CRC\_INT\_OFFSET\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-8	RESERVED	R	0h	
7-0	OFSTREG	R/W	0h	CRC Interrupt Offset. This register indicates the highest priority pending interrupt vector address. Reading the offset register automatically clear the respective interrupt flag. Please reference Table 1–3. for details.

### 20.6.6.8 CRC\_BUSY Register (Offset = 38h) [reset = 0h]

CRC\_BUSY is shown in [Figure 20-142](#) and described in [Table 20-162](#).

Return to the [Table 20-154](#).

Contains the busy flag for each channel

**Figure 20-142. CRC\_BUSY Register**

31	30	29	28	27	26	25	24
RESERVED							NU40
R-0h							R-0h
23	22	21	20	19	18	17	16
RESERVED							NU39
R-0h							R-0h
15	14	13	12	11	10	9	8
RESERVED							Ch2_BUSY
R-0h							R-0h
7	6	5	4	3	2	1	0
RESERVED							CH1_BUSY
R-0h							R-0h

**Table 20-162. CRC\_BUSY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	RESERVED	R	0h	
24	NU40	R	0h	Reserved
23-17	RESERVED	R	0h	
16	NU39	R	0h	Reserved
15-9	RESERVED	R	0h	
8	Ch2_BUSY	R	0h	Ch2_BUSY. During AUTO mode, the busy flag is set when the first data pattern of the block is compressed and remains set until the the last data pattern of the block is compressed. The flag is cleared when the last data pattern of the block is compressed.
7-1	RESERVED	R	0h	
0	CH1_BUSY	R	0h	CH1_BUSY. During AUTO mode, the busy flag is set when the first data pattern of the block is compressed and remains set until the the last data pattern of the block is compressed. The flag is cleared when the last data pattern of the block is compressed.

### 20.6.6.9 CRC\_PCOUNT\_REG1 Register (Offset = 40h) [reset = 0h]

CRC\_PCOUNT\_REG1 is shown in [Figure 20-143](#) and described in [Table 20-163](#).

Return to the [Table 20-154](#).

Channel 1 preload register for the pattern count

**Figure 20-143. CRC\_PCOUNT\_REG1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CRC_PAT_COUNT1																			
R-0h												R/W-0h																			

**Table 20-163. CRC\_PCOUNT\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	CRC_PAT_COUNT1	R/W	0h	Channel 1 Pattern Counter Preload Register. This register contains the number of data patterns in one sector to be compressed before a CRC is performed.

### 20.6.6.10 CRC\_SCOUNT\_REG1 Register (Offset = 44h) [reset = 0h]

CRC\_SCOUNT\_REG1 is shown in [Figure 20-144](#) and described in [Table 20-164](#).

Return to the [Table 20-154](#).

Channel 1 preload register for the sector count

**Figure 20-144. CRC\_SCOUNT\_REG1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_SEC_COUNT1															
R-0h																R/W-0h															

**Table 20-164. CRC\_SCOUNT\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_SEC_COUNT1	R/W	0h	Channel 1 Sector Counter Preload Register. This register contains the number of sectors in one block of memory.



### 20.6.6.11 CRC\_CURSEC\_REG1 Register (Offset = 48h) [reset = 0h]

CRC\_CURSEC\_REG1 is shown in [Figure 20-145](#) and described in [Table 20-165](#).

Return to the [Table 20-154](#).

Channel 1 current sector register contains the sector number which causes CRC failure

**Figure 20-145. CRC\_CURSEC\_REG1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_CURSEC1															
R-0h																R/W-0h															

**Table 20-165. CRC\_CURSEC\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_CURSEC1	R/W	0h	Channel 1 Current Sector ID Register. In AUTO mode, this register contains the current sector number of which the signature verification fails. The sector counter is a free running up counter. When a sector fails, the erroneous sector number is logged into current sector ID register and the CRC fail interrupt is generated. The sector ID register is frozen until it is read and the CRC fail status bit is cleared by CPU. While it is frozen, it does not capture another erroneous sector number. When this condition happens, an overrun interrupt is generated instead. Once the register is read and the CRC fail interrupt flag is cleared it can capture new erroneous sector number.

### 20.6.6.12 CRC\_WDTPD1 Register (Offset = 4Ch) [reset = 0h]

CRC\_WDTPD1 is shown in [Figure 20-146](#) and described in [Table 20-166](#).

Return to the [Table 20-154](#).

Channel 1 timeout pre-load value to check if within a given time DMA initiates a block transfer

**Figure 20-146. CRC\_WDTPD1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_WDTPD1																							
R-0h								R/W-0h																							

**Table 20-166. CRC\_WDTPD1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_WDTPD1	R/W	0h	Channel 1 Watchdog Timeout Counter Preload Register. This register contains the number of clock cycles within which the DMA must transfer the next block of data patterns.

### 20.6.6.13 CRC\_BCTOPLD1 Register (Offset = 50h) [reset = 0h]

CRC\_BCTOPLD1 is shown in [Figure 20-147](#) and described in [Table 20-167](#).

Return to the [Table 20-154](#).

Channel 1 timeout pre-load value to check if one block of patterns are compressed with a given time

**Figure 20-147. CRC\_BCTOPLD1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_BCTOPLD1																							
R-0h								R/W-0h																							

**Table 20-167. CRC\_BCTOPLD1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_BCTOPLD1	R/W	0h	Channel 1 Block Complete Timeout Counter Preload Register. This register contains the number of clock cycles within which the CRC for an entire block needs to complete before a timeout interrupt is generated.

### 20.6.6.14 PSA\_SIGREGL1 Register (Offset = 60h) [reset = 0h]

PSA\_SIGREGL1 is shown in [Figure 20-148](#) and described in [Table 20-168](#).

Return to the [Table 20-154](#).

Channel 1 PSA signature low register

**Figure 20-148. PSA\_SIGREGL1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASIG1_31_0																															
R/W-0h																															

**Table 20-168. PSA\_SIGREGL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSASIG1_31_0	R/W	0h	Channel 1 PSA Signature Low Register. This register contains the value stored at PSASIG1[31:0] register.

### 20.6.6.15 PSA\_SIGREGH1 Register (Offset = 64h) [reset = 0h]

PSA\_SIGREGH1 is shown in [Figure 20-149](#) and described in [Table 20-169](#).

Return to the [Table 20-154](#).

Channel 1 PSA signature high register

**Figure 20-149. PSA\_SIGREGH1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSA_SIG1_63_32																															
R/W-0h																															

**Table 20-169. PSA\_SIGREGH1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSA_SIG1_63_32	R/W	0h	Channel 1 PSA Signature High Register. This register contains the value stored at PSASIG1[63:32] register.

### 20.6.6.16 CRC\_REGL1 Register (Offset = 68h) [reset = 0h]

CRC\_REGL1 is shown in [Figure 20-150](#) and described in [Table 20-170](#).

Return to the [Table 20-154](#).

Channel 1 CRC value low register

**Figure 20-150. CRC\_REGL1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC1_31_0																															
R/W-0h																															

**Table 20-170. CRC\_REGL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CRC1_31_0	R/W	0h	Channel 1 CRC Value Low Register. This register contains the current known good signature value stored at CRC1[31:0] register.

### 20.6.6.17 CRC\_REGH1 Register (Offset = 6Ch) [reset = 0h]

CRC\_REGH1 is shown in [Figure 20-151](#) and described in [Table 20-171](#).

Return to the [Table 20-154](#).

Channel 1 CRC value high register

**Figure 20-151. CRC\_REGH1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC1_63_32																															
R/W-0h																															

**Table 20-171. CRC\_REGH1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CRC1_63_32	R/W	0h	Channel 1 CRC Value High Register. This register contains the current known good signature value stored at CRC1[63:32] register.

### 20.6.6.18 PSA\_SECSIGREGL1 Register (Offset = 70h) [reset = 0h]

PSA\_SECSIGREGL1 is shown in [Figure 20-152](#) and described in [Table 20-172](#).

Return to the [Table 20-154](#).

Channel 1 PSA sector signature low regis-ter

**Figure 20-152. PSA\_SECSIGREGL1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG1_31_0																															
R-0h																															

**Table 20-172. PSA\_SECSIGREGL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSASECSIG1_31_0	R	0h	Channel 1 PSA Sector Signature Low Register. This register contains the value stored at PSASECSIG1[31:0] register.



### 20.6.6.19 PSA\_SECSIGREGH1 Register (Offset = 74h) [reset = 0h]

PSA\_SECSIGREGH1 is shown in [Figure 20-153](#) and described in [Table 20-173](#).

Return to the [Table 20-154](#).

Channel 1 PSA sector signature high register

**Figure 20-153. PSA\_SECSIGREGH1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG1_63_32																															
R-0h																															

**Table 20-173. PSA\_SECSIGREGH1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSASECSIG1_63_32	R	0h	Channel 1 PSA Sector Signature High Register. This register contains the value stored at PSASECSIG1[63:32] register.

### 20.6.6.20 RAW\_DATAREGL1 Register (Offset = 78h) [reset = 0h]

RAW\_DATAREGL1 is shown in [Figure 20-154](#) and described in [Table 20-174](#).

Return to the [Table 20-154](#).

Channel 1 un-compressed raw data low register

**Figure 20-154. RAW\_DATAREGL1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA1_31_0																															
R-0h																															

**Table 20-174. RAW\_DATAREGL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RAW_DATA1_31_0	R	0h	Channel 1 Raw Data Low Register. This register contains bit 31:0 of the un-compressed raw data.

### 20.6.6.21 RAW\_DATAREGH1 Register (Offset = 7Ch) [reset = 0h]

RAW\_DATAREGH1 is shown in [Figure 20-155](#) and described in [Table 20-175](#).

Return to the [Table 20-154](#).

Channel 1 un-compressed raw data high register

**Figure 20-155. RAW\_DATAREGH1 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA1_63_32																															
R-0h																															

**Table 20-175. RAW\_DATAREGH1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RAW_DATA1_63_32	R	0h	Channel 1 Raw Data High Register. This register contains bit 63:32 of the un-compressed raw data.

### 20.6.6.22 CRC\_PCOUNT\_REG2 Register (Offset = 80h) [reset = 0h]

CRC\_PCOUNT\_REG2 is shown in [Figure 20-156](#) and described in [Table 20-176](#).

Return to the [Table 20-154](#).

Channel 2 preload register for the pattern count

**Figure 20-156. CRC\_PCOUNT\_REG2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CRC_PAT_COUNT2																			
R-0h												R/W-0h																			

**Table 20-176. CRC\_PCOUNT\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	CRC_PAT_COUNT2	R/W	0h	Channel 2 Pattern Counter Preload Register. This register contains the number of data patterns in one sector to be compressed before a CRC is performed.

### 20.6.6.23 CRC\_SCOUNT\_REG2 Register (Offset = 84h) [reset = 0h]

CRC\_SCOUNT\_REG2 is shown in [Figure 20-157](#) and described in [Table 20-177](#).

Return to the [Table 20-154](#).

Channel 2 preload register for the sector count

**Figure 20-157. CRC\_SCOUNT\_REG2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_SEC_COUNT2															
R-0h																R/W-0h															

**Table 20-177. CRC\_SCOUNT\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_SEC_COUNT2	R/W	0h	Channel 2 Sector Counter Preload Register. This register contains the number of sectors in one block of memory.

### 20.6.6.24 CRC\_CURSEC\_REG2 Register (Offset = 88h) [reset = 0h]

CRC\_CURSEC\_REG2 is shown in [Figure 20-158](#) and described in [Table 20-178](#).

Return to the [Table 20-154](#).

Channel 2 current sector register contains the sector number which causes CRC fail-ure

**Figure 20-158. CRC\_CURSEC\_REG2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CRC_CURSEC2															
R-0h																R/W-0h															

**Table 20-178. CRC\_CURSEC\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	CRC_CURSEC2	R/W	0h	Channel 2 Current Sector ID Register. In AUTO mode, this register contains the current sector number of which the signature verification fails. The sector counter is a free running up counter. When a sector fails, the erroneous sector number is logged into current sector ID register and the CRC fail interrupt is generated. The sector ID register is frozen until it is read and the CRC fail status bit is cleared by CPU. While it is frozen, it does not capture another erroneous sector number. When this condition happens, an overrun interrupt is generated instead. Once the register is read and the CRC fail interrupt flag is cleared it can capture new erroneous sector number.

### 20.6.6.25 CRC\_WDTPD2 Register (Offset = 8Ch) [reset = 0h]

CRC\_WDTPD2 is shown in [Figure 20-159](#) and described in [Table 20-179](#).

Return to the [Table 20-154](#).

Channel 2 timeout pre-load value to check if within a given time DMA initiates a block transfer

**Figure 20-159. CRC\_WDTPD2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_WDTPD2																							
R-0h								R/W-0h																							

**Table 20-179. CRC\_WDTPD2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_WDTPD2	R/W	0h	Channel 2 Watchdog Timeout Counter Preload Register. This register contains the number of clock cycles within which the DMA must transfer the next block of data patterns.

### 20.6.6.26 CRC\_BCTOPLD2 Register (Offset = 90h) [reset = 0h]

CRC\_BCTOPLD2 is shown in [Figure 20-160](#) and described in [Table 20-180](#).

Return to the [Table 20-154](#).

Channel 2 timeout pre-load value to check if one block of patterns are compressed with a given time

**Figure 20-160. CRC\_BCTOPLD2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRC_BCTOPLD2																							
R-0h								R/W-0h																							

**Table 20-180. CRC\_BCTOPLD2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	CRC_BCTOPLD2	R/W	0h	Channel 2 Block Complete Timeout Counter Preload Register. This register contains the number of clock cycles within which the CRC for an entire block needs to complete before a timeout interrupt is generated.



### 20.6.6.27 PSA\_SIGREGL2 Register (Offset = A0h) [reset = 0h]

PSA\_SIGREGL2 is shown in [Figure 20-161](#) and described in [Table 20-181](#).

Return to the [Table 20-154](#).

Channel 2 PSA signature low register

**Figure 20-161. PSA\_SIGREGL2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASIG2_31_0																															
R/W-0h																															

**Table 20-181. PSA\_SIGREGL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSASIG2_31_0	R/W	0h	Channel 2 PSA Signature Low Register. This register contains the value stored at PSASIG2[31:0] register.

### 20.6.6.28 PSA\_SIGREGH2 Register (Offset = A4h) [reset = 0h]

PSA\_SIGREGH2 is shown in [Figure 20-162](#) and described in [Table 20-182](#).

Return to the [Table 20-154](#).

Channel 2 PSA signature high register

**Figure 20-162. PSA\_SIGREGH2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSA_SIG2_63_32																															
R/W-0h																															

**Table 20-182. PSA\_SIGREGH2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSA_SIG2_63_32	R/W	0h	Channel 2 PSA Signature High Register. This register contains the value stored at PSASIG2[63:32] register.

### 20.6.6.29 CRC\_REGL2 Register (Offset = A8h) [reset = 0h]

CRC\_REGL2 is shown in [Figure 20-163](#) and described in [Table 20-183](#).

Return to the [Table 20-154](#).

Channel 2 CRC value low register

**Figure 20-163. CRC\_REGL2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC2_31_0																															
R/W-0h																															

**Table 20-183. CRC\_REGL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CRC2_31_0	R/W	0h	Channel 2 CRC Value Low Register. This register contains the current known good signature value stored at CRC2[31:0] register.

### 20.6.6.30 CRC\_REGH2 Register (Offset = ACh) [reset = 0h]

CRC\_REGH2 is shown in [Figure 20-164](#) and described in [Table 20-184](#).

Return to the [Table 20-154](#).

Channel 2 CRC value high register

**Figure 20-164. CRC\_REGH2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CRC2_63_32																															
R/W-0h																															

**Table 20-184. CRC\_REGH2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CRC2_63_32	R/W	0h	Channel 2 CRC Value High Register. This register contains the current known good signature value stored at CRC2[63:32] register.

### 20.6.6.31 PSA\_SECSIGREGL2 Register (Offset = B0h) [reset = 0h]

PSA\_SECSIGREGL2 is shown in [Figure 20-165](#) and described in [Table 20-185](#).

Return to the [Table 20-154](#).

Channel 2 PSA sector signature low regis-ter

**Figure 20-165. PSA\_SECSIGREGL2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG2_31_0																															
R-0h																															

**Table 20-185. PSA\_SECSIGREGL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSASECSIG2_31_0	R	0h	Channel 2 PSA Sector Signature Low Register. This register contains the value stored at PSASECSIG2[31:0] register.

### 20.6.6.32 PSA\_SECSIGREGH2 Register (Offset = B4h) [reset = 0h]

PSA\_SECSIGREGH2 is shown in [Figure 20-166](#) and described in [Table 20-186](#).

Return to the [Table 20-154](#).

Channel 2 PSA sector signature high register

**Figure 20-166. PSA\_SECSIGREGH2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PSASECSIG2_63_32																															
R-0h																															

**Table 20-186. PSA\_SECSIGREGH2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PSASECSIG2_63_32	R	0h	Channel 2 PSA Sector Signature High Register. This register contains the value stored at PSASECSIG2[63:32] register.

### 20.6.6.33 RAW\_DATAREGL2 Register (Offset = B8h) [reset = 0h]

RAW\_DATAREGL2 is shown in [Figure 20-167](#) and described in [Table 20-187](#).

Return to the [Table 20-154](#).

Channel 2 un-compressed raw data low register

**Figure 20-167. RAW\_DATAREGL2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA2_31_0																															
R-0h																															

**Table 20-187. RAW\_DATAREGL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RAW_DATA2_31_0	R	0h	Channel 2 Raw Data Low Register. This register contains bit 31:0 of the un-compressed raw data.

### 20.6.6.34 RAW\_DATAREGH2 Register (Offset = BCh) [reset = 0h]

RAW\_DATAREGH2 is shown in [Figure 20-168](#) and described in [Table 20-188](#).

Return to the [Table 20-154](#).

Channel 2 un-compressed raw data high Register

**Figure 20-168. RAW\_DATAREGH2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RAW_DATA2_63_32																															
R-0h																															

**Table 20-188. RAW\_DATAREGH2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	RAW_DATA2_63_32	R	0h	Channel 2 Raw Data High Register. This register contains bit 63:32 of the un-compressed raw data.



### 20.6.6.35 CRC\_PCOUNT\_REG3 Register (Offset = C0h) [reset = 0h]

CRC\_PCOUNT\_REG3 is shown in [Figure 20-169](#) and described in [Table 20-189](#).

Return to the [Table 20-154](#).

Channel 3 preload register for the pattern count

**Figure 20-169. CRC\_PCOUNT\_REG3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												NU41																			
R-0h												R-0h																			

**Table 20-189. CRC\_PCOUNT\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	NU41	R	0h	Reserved

### 20.6.6.36 CRC\_SCOUNT\_REG3 Register (Offset = C4h) [reset = 0h]

CRC\_SCOUNT\_REG3 is shown in [Figure 20-170](#) and described in [Table 20-190](#).

Return to the [Table 20-154](#).

Channel 3 preload register for the sector count

**Figure 20-170. CRC\_SCOUNT\_REG3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NU42															
R-0h																R-0h															

**Table 20-190. CRC\_SCOUNT\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU42	R	0h	Reserved

### 20.6.6.37 CRC\_CURSEC\_REG3 Register (Offset = C8h) [reset = 0h]

CRC\_CURSEC\_REG3 is shown in [Figure 20-171](#) and described in [Table 20-191](#).

Return to the [Table 20-154](#).

Channel 3 current sector register contains the sector number which causes CRC fail-ure

**Figure 20-171. CRC\_CURSEC\_REG3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NU43															
R-0h																R-0h															

**Table 20-191. CRC\_CURSEC\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU43	R	0h	Reserved

### 20.6.6.38 CRC\_WDTPLD3 Register (Offset = CCh) [reset = 0h]

CRC\_WDTPLD3 is shown in [Figure 20-172](#) and described in [Table 20-192](#).

Return to the [Table 20-154](#).

Channel 3 timeout pre-load value to check if within a given time DMA initiates a block transfer

**Figure 20-172. CRC\_WDTPLD3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NU44																							
R-0h								R-0h																							

**Table 20-192. CRC\_WDTPLD3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU44	R	0h	Reserved

### 20.6.6.39 CRC\_BCTOPLD3 Register (Offset = D0h) [reset = 0h]

CRC\_BCTOPLD3 is shown in [Figure 20-173](#) and described in [Table 20-193](#).

Return to the [Table 20-154](#).

Channel 3 timeout pre-load value to check if one block of patterns are compressed with a given time

**Figure 20-173. CRC\_BCTOPLD3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NU45																							
R-0h								R-0h																							

**Table 20-193. CRC\_BCTOPLD3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU45	R	0h	Reserved

#### 20.6.6.40 PSA\_SIGREGL3 Register (Offset = E0h) [reset = 0h]

PSA\_SIGREGL3 is shown in [Figure 20-174](#) and described in [Table 20-194](#).

Return to the [Table 20-154](#).

Channel 3 PSA signature low register

**Figure 20-174. PSA\_SIGREGL3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU46																															
R-0h																															

**Table 20-194. PSA\_SIGREGL3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU46	R	0h	Reserved

#### 20.6.6.41 PSA\_SIGREGH3 Register (Offset = E4h) [reset = 0h]

PSA\_SIGREGH3 is shown in [Figure 20-175](#) and described in [Table 20-195](#).

Return to the [Table 20-154](#).

Channel 3 PSA signature high register

**Figure 20-175. PSA\_SIGREGH3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU47																															
R-0h																															

**Table 20-195. PSA\_SIGREGH3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU47	R	0h	Reserved

### 20.6.6.42 CRC\_REGL3 Register (Offset = E8h) [reset = 0h]

CRC\_REGL3 is shown in [Figure 20-176](#) and described in [Table 20-196](#).

Return to the [Table 20-154](#).

Channel 3 CRC value low register

**Figure 20-176. CRC\_REGL3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU48																															
R-0h																															

**Table 20-196. CRC\_REGL3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU48	R	0h	Reserved



### 20.6.6.43 CRC\_REGH3 Register (Offset = ECh) [reset = 0h]

CRC\_REGH3 is shown in [Figure 20-177](#) and described in [Table 20-197](#).

Return to the [Table 20-154](#).

Channel 3 CRC value high register

**Figure 20-177. CRC\_REGH3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU49																															
R-0h																															

**Table 20-197. CRC\_REGH3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU49	R	0h	Reserved

#### 20.6.6.44 PSA\_SECSIGREGL3 Register (Offset = F0h) [reset = 0h]

PSA\_SECSIGREGL3 is shown in [Figure 20-178](#) and described in [Table 20-198](#).

Return to the [Table 20-154](#).

Channel 3 PSA sector signature low regis-ter

**Figure 20-178. PSA\_SECSIGREGL3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU50																															
R-0h																															

**Table 20-198. PSA\_SECSIGREGL3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU50	R	0h	Reserved

### 20.6.6.45 PSA\_SECSIGREGH3 Register (Offset = F4h) [reset = 0h]

PSA\_SECSIGREGH3 is shown in [Figure 20-179](#) and described in [Table 20-199](#).

Return to the [Table 20-154](#).

Channel 3 PSA sector signature high register

**Figure 20-179. PSA\_SECSIGREGH3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU51																															
R-0h																															

**Table 20-199. PSA\_SECSIGREGH3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU51	R	0h	Reserved

### 20.6.6.46 RAW\_DATAREGL3 Register (Offset = F8h) [reset = 0h]

RAW\_DATAREGL3 is shown in [Figure 20-180](#) and described in [Table 20-200](#).

Return to the [Table 20-154](#).

Channel 3 un-compressed raw data low register

**Figure 20-180. RAW\_DATAREGL3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU52																															
R-0h																															

**Table 20-200. RAW\_DATAREGL3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU52	R	0h	Reserved

### 20.6.6.47 RAW\_DATAAREGH3 Register (Offset = FCh) [reset = 0h]

RAW\_DATAAREGH3 is shown in [Figure 20-181](#) and described in [Table 20-201](#).

Return to the [Table 20-154](#).

Channel 3 un-compressed raw data high Register

**Figure 20-181. RAW\_DATAAREGH3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU53																															
R-0h																															

**Table 20-201. RAW\_DATAAREGH3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU53	R	0h	Reserved

### 20.6.6.48 CRC\_PCOUNT\_REG4 Register (Offset = 100h) [reset = 0h]

CRC\_PCOUNT\_REG4 is shown in [Figure 20-182](#) and described in [Table 20-202](#).

Return to the [Table 20-154](#).

Channel 4 preload register for the pattern count

**Figure 20-182. CRC\_PCOUNT\_REG4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												NU54																			
R-0h												R-0h																			

**Table 20-202. CRC\_PCOUNT\_REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	RESERVED	R	0h	
19-0	NU54	R	0h	Reserved

### 20.6.6.49 CRC\_SCOUNT\_REG4 Register (Offset = 104h) [reset = 0h]

CRC\_SCOUNT\_REG4 is shown in [Figure 20-183](#) and described in [Table 20-203](#).

Return to the [Table 20-154](#).

Channel 4 preload register for the sector count

**Figure 20-183. CRC\_SCOUNT\_REG4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NU55															
R-0h																R-0h															

**Table 20-203. CRC\_SCOUNT\_REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU55	R	0h	Reserved

### 20.6.6.50 CRC\_CURSEC\_REG4 Register (Offset = 108h) [reset = 0h]

CRC\_CURSEC\_REG4 is shown in [Figure 20-184](#) and described in [Table 20-204](#).

Return to the [Table 20-154](#).

Channel 4 current sector register contains the sector number which causes CRC fail-ure

**Figure 20-184. CRC\_CURSEC\_REG4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NU56															
R-0h																R-0h															

**Table 20-204. CRC\_CURSEC\_REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	RESERVED	R	0h	
15-0	NU56	R	0h	Reserved



### 20.6.6.51 CRC\_WDTPD4 Register (Offset = 10Ch) [reset = 0h]

CRC\_WDTPD4 is shown in [Figure 20-185](#) and described in [Table 20-205](#).

Return to the [Table 20-154](#).

Channel 4 timeout pre-load value to check if within a given time DMA initiates a block transfer

**Figure 20-185. CRC\_WDTPD4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NU57																							
R-0h								R-0h																							

**Table 20-205. CRC\_WDTPD4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU57	R	0h	Reserved

### 20.6.6.52 CRC\_BCTOPLD4 Register (Offset = 110h) [reset = 0h]

CRC\_BCTOPLD4 is shown in [Figure 20-186](#) and described in [Table 20-206](#).

Return to the [Table 20-154](#).

Channel 4 timeout pre-load value to check if one block of patterns are compressed with a given time

**Figure 20-186. CRC\_BCTOPLD4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								NU58																							
R-0h								R-0h																							

**Table 20-206. CRC\_BCTOPLD4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	RESERVED	R	0h	
23-0	NU58	R	0h	Reserved

### 20.6.6.53 PSA\_SIGREGL4 Register (Offset = 120h) [reset = 0h]

PSA\_SIGREGL4 is shown in [Figure 20-187](#) and described in [Table 20-207](#).

Return to the [Table 20-154](#).

Channel 4 PSA signature low register

**Figure 20-187. PSA\_SIGREGL4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU59																															
R-0h																															

**Table 20-207. PSA\_SIGREGL4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU59	R	0h	Reserved

### 20.6.6.54 PSA\_SIGREGH4 Register (Offset = 124h) [reset = 0h]

PSA\_SIGREGH4 is shown in [Figure 20-188](#) and described in [Table 20-208](#).

Return to the [Table 20-154](#).

Channel 4 PSA signature high register

**Figure 20-188. PSA\_SIGREGH4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU60																															
R-0h																															

**Table 20-208. PSA\_SIGREGH4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU60	R	0h	Reserved

### 20.6.6.55 CRC\_REGL4 Register (Offset = 128h) [reset = 0h]

CRC\_REGL4 is shown in [Figure 20-189](#) and described in [Table 20-209](#).

Return to the [Table 20-154](#).

Channel 4 CRC value low register

**Figure 20-189. CRC\_REGL4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU61																															
R-0h																															

**Table 20-209. CRC\_REGL4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU61	R	0h	Reserved

### 20.6.6.56 CRC\_REGH4 Register (Offset = 12Ch) [reset = 0h]

CRC\_REGH4 is shown in [Figure 20-190](#) and described in [Table 20-210](#).

Return to the [Table 20-154](#).

Channel 4 CRC value high register

**Figure 20-190. CRC\_REGH4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU62																															
R-0h																															

**Table 20-210. CRC\_REGH4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU62	R	0h	Reserved

### 20.6.6.57 PSA\_SECSIGREGL4 Register (Offset = 130h) [reset = 0h]

PSA\_SECSIGREGL4 is shown in [Figure 20-191](#) and described in [Table 20-211](#).

Return to the [Table 20-154](#).

Channel 4 PSA sector signature low regis-ter

**Figure 20-191. PSA\_SECSIGREGL4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU63																															
R-0h																															

**Table 20-211. PSA\_SECSIGREGL4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU63	R	0h	Reserved

### 20.6.6.58 PSA\_SECSIGREGH4 Register (Offset = 134h) [reset = 0h]

PSA\_SECSIGREGH4 is shown in [Figure 20-192](#) and described in [Table 20-212](#).

Return to the [Table 20-154](#).

Channel 4 PSA sector signature high register

**Figure 20-192. PSA\_SECSIGREGH4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU64																															
R-0h																															

**Table 20-212. PSA\_SECSIGREGH4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU64	R	0h	Reserved



### 20.6.6.59 RAW\_DATAREGL4 Register (Offset = 138h) [reset = 0h]

RAW\_DATAREGL4 is shown in [Figure 20-193](#) and described in [Table 20-213](#).

Return to the [Table 20-154](#).

Channel 4 un-compressed raw data low register

**Figure 20-193. RAW\_DATAREGL4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU65																															
R-0h																															

**Table 20-213. RAW\_DATAREGL4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU65	R	0h	Reserved

### 20.6.6.60 RAW\_DATAREGH4 Register (Offset = 13Ch) [reset = 0h]

RAW\_DATAREGH4 is shown in [Figure 20-194](#) and described in [Table 20-214](#).

Return to the [Table 20-154](#).

Channel 4 un-compressed raw data high Register

**Figure 20-194. RAW\_DATAREGH4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU66																															
R-0h																															

**Table 20-214. RAW\_DATAREGH4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU66	R	0h	Reserved

### 20.6.6.61 MCRC\_BUS\_SEL Register (Offset = 140h) [reset = 7h]

MCRC\_BUS\_SEL is shown in [Figure 20-195](#) and described in [Table 20-215](#).

Return to the [Table 20-154](#).

Disables either or all tracing of data buses

**Figure 20-195. MCRC\_BUS\_SEL Register**

31	30	29	28	27	26	25	24
NU67							
R-0h							
23	22	21	20	19	18	17	16
NU67							
R-0h							
15	14	13	12	11	10	9	8
NU67							
R-0h							
7	6	5	4	3	2	1	0
NU67					MEn	DTCMEn	ITCMEn
R-0h					R/W-1h	R/W-1h	R/W-1h

**Table 20-215. MCRC\_BUS\_SEL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-3	NU67	R	0h	Reserved
2	MEn	R/W	1h	MEn. Enable/disables the tracing of VBUSM 0: Tracing of VBUSM controller bus has been disabled 1: Tracing of VBUSM controller bus has been enabled
1	DTCMEn	R/W	1h	DTCMEn. Enable/disables the tracing of data TCM 0: Tracing of DTCM_ODD and DTCM_EVEN buses have been disabled 1: Tracing of DTCM_ODD and DTCM_EVEN buses have been enabled
0	ITCMEn	R/W	1h	ITCMEn. Enable/disables the tracing of instruction TCM 0: Tracing of ITCM bus has been disabled 1: Tracing of ITCM bus has been enabled

### 20.6.6.62 MCRC\_RESERVED Register (Offset = 144h) [reset = 0h]

MCRC\_RESERVED is shown in [Figure 20-196](#) and described in [Table 20-216](#).

Return to the [Table 20-154](#).

0x144 to 0x1FF is reserved area.

**Figure 20-196. MCRC\_RESERVED Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU68																															
R-0h																															

**Table 20-216. MCRC\_RESERVED Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	NU68	R	0h	0x144 to 0x1FF is reserved area.

## 20.7 Self-Test Controller (STC)

### 20.7.1 Integration Spec

#### 20.7.1.1 STC Memory Map

**Table 20-217. STC Memory Map AWR2544**

Name	Start Address	Frame Address (Hex) End	Size	Description
MSS_R5SS_STC	0x02F7 9800	0x02F7 9918	284 Bytes	MSS_STC module configuration registers

#### 20.7.1.2 Unsupported Features

- [Section 20.7.4.7.1](#) – Launch-on-last-shift. TR\_T =1
- [Section 20.7.4.7.2](#) – Transition delay fault model. FT =1
- [Section 20.7.4.7.6](#) – Low-power scan mode. MSS\_STC.STCGCR1.LP\_SCAN\_MODE = 1
- [Section 20.7.4.7.7](#) and [Section 20.7.4.7.8](#) – Coverage improvement techniques – MSS\_STC.STCGCR1.ROM\_ACCESS\_INV =1 Mode
- Interval-based testing
- MSS\_STC.STC\_CLKDIV clock division features

### 20.7.2 STC Overview

The enhanced Self-Test Controller (STC) is used to test logic cores based on the On-Product Multiple Input Signature Register (OPMISR) scan compression architecture.

Software-based self-test programs for the cores are available, but offer less test coverage. Due to the complexity of the soft cores, the coverage required can be difficult to achieve and will result in a larger program size.

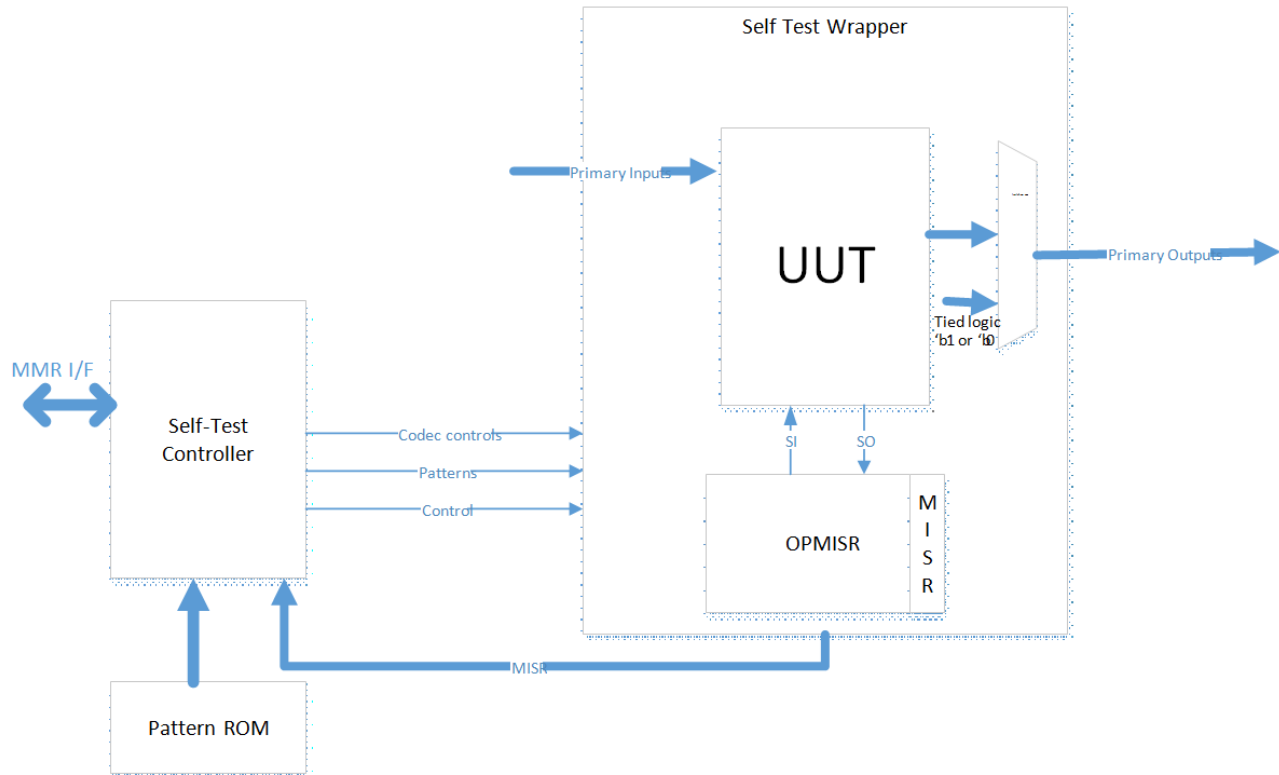
For these complex cores, on-chip logic BIST support for the self-test is preferred.

The main features of the STC include:

- Implements the OPMISR controller, along with the on-chip self-test controller for the synthesizable module logic, which enables high test coverage.
- The self-test controller facilitates complete isolation of the logical segment under the test from the rest of the system during the self-test run. Configure critical control signals in the initiator and target ports of the logical segment under the test to a safe state.
- The self-tested CPU core initiator bus transaction signals are configured to be in idle mode during the self-test run.
- Time-out counter for the self-test run as a fail-safe feature.
- Can capture power reduction using dead cycles before and after the capture pulse.

A self test segment corresponds to a portion of discreet safety-critical logic which can be tested in isolation from the rest of the system by the self test controller and OPMISR logic.

### 20.7.2.1 OPMISR Concept



**Figure 20-197. OPMISR Conceptual Diagram**

The On-Product Multiple-Input Signature Register (OPMISR) is a methodology which moves the test pattern generation on-chip. Logic BIST is implemented on functional partitions (BIST'ed COREs) that are speed-critical and have high gate count. A conceptual diagram of OPMISR implementation is shown in [Figure 20-197](#).

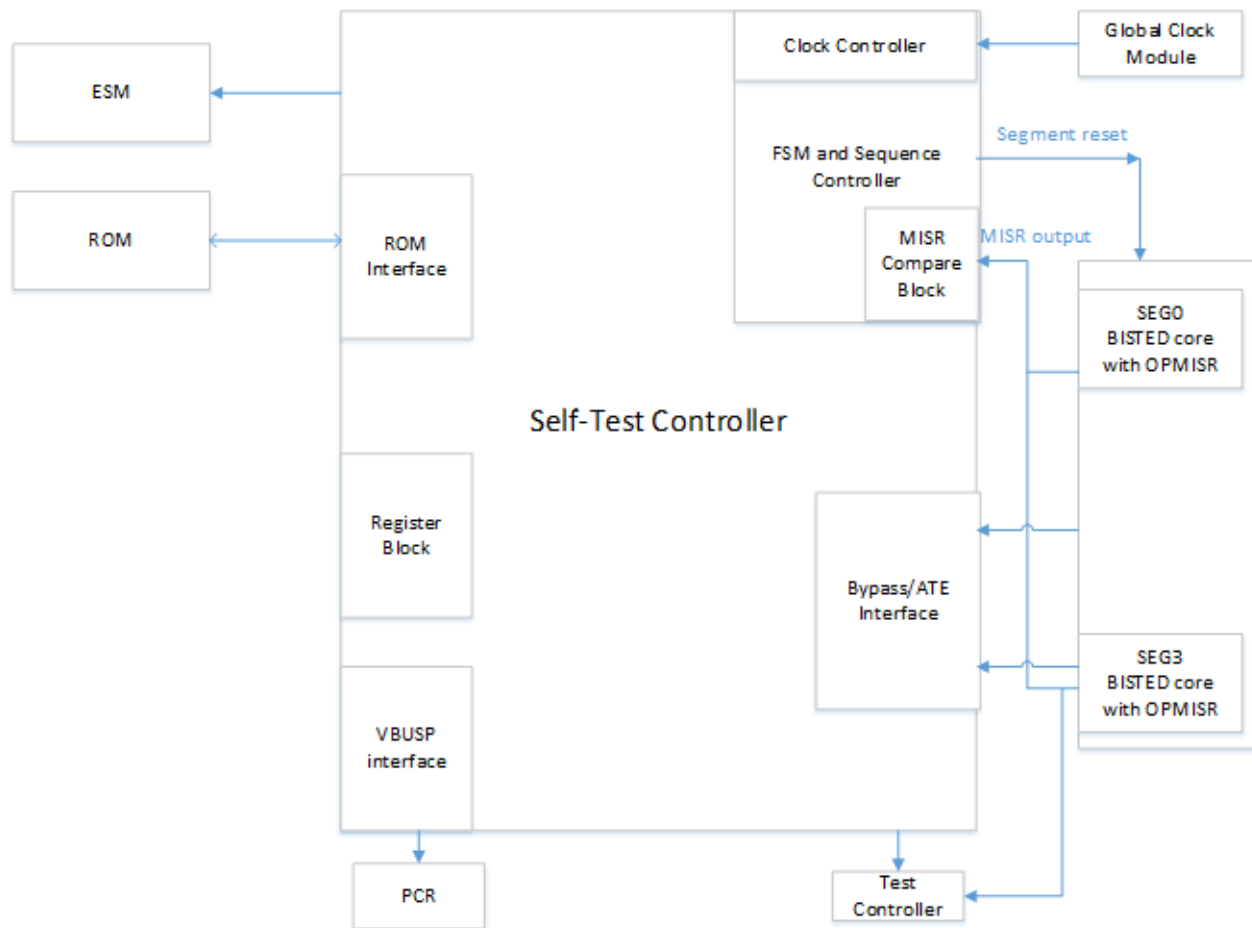
The MISR test structure modifies the typical fullscan scan chain such that each scan data input internally drives many chains. These chains feed to the inserted MISR structure. The chain's values are captured into the MISR during shift, generating a resulting signature that can be shifted out.

A given Unit Under Test (UUT) is scan-inserted, and the scan chains are hooked to the OPMISR logic. The self-test wrapper created around the UUT and the OPMISR logic includes the isolation muxes for the output ports of the core, to ensure that the core and UUT are isolated from the rest of the system during the self-test.

### 20.7.3 Block Diagram

The STC module is composed of following blocks:

- ROM interface
- FSM and sequence control
- Register file
- STC bypass / ATE interface
- Peripheral bus interface (VBUSP interface)



#### **20.7.4.3 Register Block**

This block implements the user-programmable control registers that determine when to start a self test and what clock frequency the scan test should be performed.

The register block also captures various status information of the self test for the user.

#### **20.7.4.4 STC Bypass / ATE Interface**

This is a production test interface. This section bypasses the self-test FSM. The OPMISR signal interface is brought out directly to the module ports, and these are accessible to the ATE (tester) at the device level using the test controller module. The intent of the block is to provide capability for fault isolation for parts failing the logic self test run.

This block receives two sets of signals; one from the device test controller, and another similar set from the self-test FSM (test sequencer). The bypass indicator signal is used to select one of the two sets of signals to be routed to the OPMISR controller.

#### **20.7.4.5 VBUSP Interface**

The control and the status registers of the STC module can be accessed through the VBUSP interface. During application programming, configuration registers are programmed through the peripheral interface, to enable and run the self-test controller.

## 20.7.4.6 STC Flow

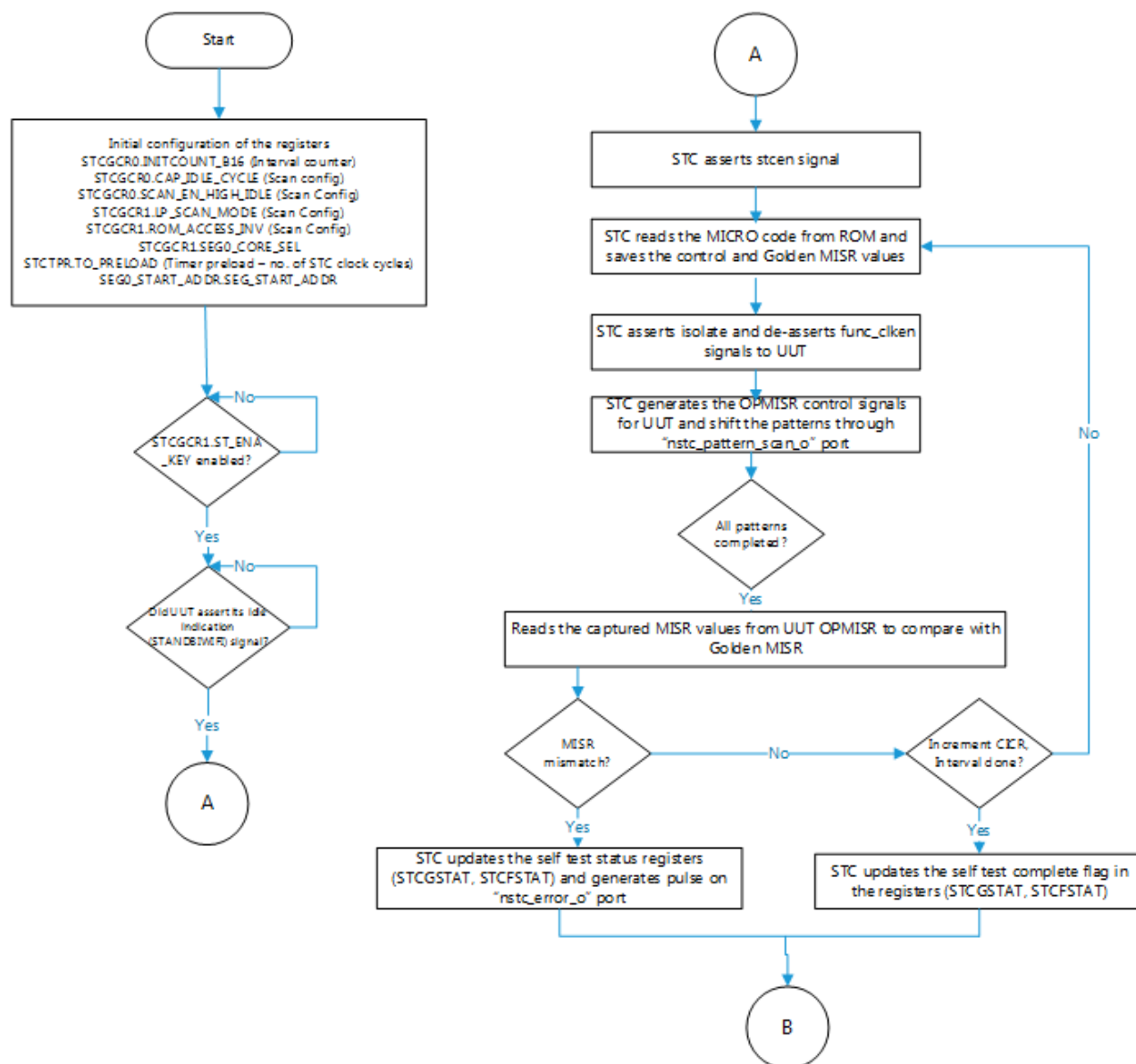
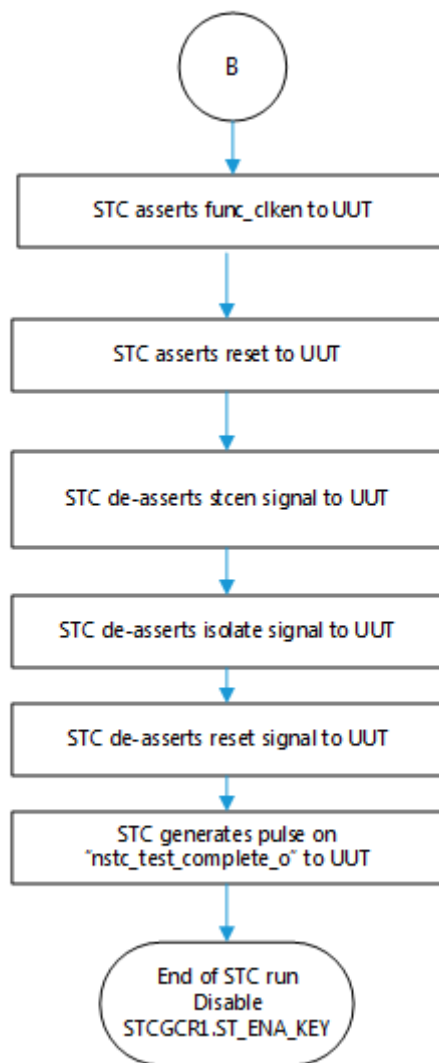


Figure 20-199. STC Flow (1 of 2)





**Figure 20-200. STC Flow (2 of 2)**

#### 20.7.4.7 ROM Organization

**Table 20-218. ROM Organization for 2 Intervals**

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
<b>INTERVAL 0</b>									
CFG for interval 0, when rom_access_inversion =0	Reserved	pattern_cou nt[9:0]	Reserved	Reserved	Reserved	Seg_ID[1]	Seg_ID[0]	FT	TR_T

**Table 20-218. ROM Organization for 2 Intervals (continued)**

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
MISR for interval 0, when rom_access_inversion =0	MISR_GOLDEN[895:840]								
	MISR_GOLDEN[839:784]								
	MISR_GOLDEN[783:728]								
	MISR_GOLDEN[727:672]								
	MISR_GOLDEN[671:616]								
	MISR_GOLDEN[615:560]								
	MISR_GOLDEN[559:504]								
	MISR_GOLDEN[503:448]								
	MISR_GOLDEN[447:392]								
	MISR_GOLDEN[391:336]								
	MISR_GOLDEN[335:280]								
	MISR_GOLDEN[279:224]								
	MISR_GOLDEN[223:168]								
	MISR_GOLDEN[167:112]								
	MISR_GOLDEN[111:56]								
	MISR_GOLDEN[55:0]								
LP_MISR for interval 0, when rom_access_inversion =0	LP_MISR_GOLDEN[895:840]								
	LP_MISR_GOLDEN[839:784]								
	LP_MISR_GOLDEN[783:728]								
	LP_MISR_GOLDEN[727:672]								
	LP_MISR_GOLDEN[671:616]								
	LP_MISR_GOLDEN[615:560]								
	LP_MISR_GOLDEN[559:504]								
	LP_MISR_GOLDEN[503:448]								
	LP_MISR_GOLDEN[447:392]								
	LP_MISR_GOLDEN[391:336]								
	LP_MISR_GOLDEN[335:280]								
	LP_MISR_GOLDEN[279:224]								
	LP_MISR_GOLDEN[223:168]								
	LP_MISR_GOLDEN[167:112]								
	LP_MISR_GOLDEN[111:56]								
	LP_MISR_GOLDEN[55:0]								
Patterns for interval 0	P1_SD8[6:0]	P1_SD7[6:0]	P1_SD6[6:0]	...	...	...	P1_SD1[6:0]		
		...	...	...	...	...	...	P1_SD9[6:0]	
		...	...	...	...	...	...	...	...

**Table 20-218. ROM Organization for 2 Intervals (continued)**

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
LP_MISR for interval 0, when rom_access_inversion =1	LP_INV_MISR_GOLDEN[55:0]								
	LP_INV_MISR_GOLDEN[111:56]								
	LP_INV_MISR_GOLDEN[167:112]								
	LP_INV_MISR_GOLDEN[223:168]								
	LP_INV_MISR_GOLDEN[279:224]								
	LP_INV_MISR_GOLDEN[335:280]								
	LP_INV_MISR_GOLDEN[391:336]								
	LP_INV_MISR_GOLDEN[447:392]								
	LP_INV_MISR_GOLDEN[503:448]								
	LP_INV_MISR_GOLDEN[559:504]								
	LP_INV_MISR_GOLDEN[615:560]								
	LP_INV_MISR_GOLDEN[671:616]								
	LP_INV_MISR_GOLDEN[727:672]								
	LP_INV_MISR_GOLDEN[783:728]								
	LP_INV_MISR_GOLDEN[839:784]								
	LP_INV_MISR_GOLDEN[895:840]								
MISR for interval 0, when rom_access_inversion =1	INV_MISR_GOLDEN[55:0]								
	INV_MISR_GOLDEN[111:56]								
	INV_MISR_GOLDEN[167:112]								
	INV_MISR_GOLDEN[223:168]								
	INV_MISR_GOLDEN[279:224]								
	INV_MISR_GOLDEN[335:280]								
	INV_MISR_GOLDEN[391:336]								
	INV_MISR_GOLDEN[447:392]								
	INV_MISR_GOLDEN[503:448]								
	INV_MISR_GOLDEN[559:504]								
	INV_MISR_GOLDEN[615:560]								
	INV_MISR_GOLDEN[671:616]								
	INV_MISR_GOLDEN[727:672]								
	INV_MISR_GOLDEN[783:728]								
	INV_MISR_GOLDEN[839:784]								
	INV_MISR_GOLDEN[895:840]								
CFG for interval 0, when rom_access_inversion =1 (same as when_rom_access_inversion =0)	Reserved	pattern_cou nt[9:0]	Reserved	Reserved	Reserved	Seg_ID[1]	Seg_ID[0 ]	FT	TR_T
<b>INTERVAL 1</b>									
CFG for interval 1, when rom_access_inversion =0	Reserved	pattern_cou nt[9:0]	Reserved	Reserved	Reserved	Seg_ID[1]	Seg_ID[0 ]	FT	TR_T

**Table 20-218. ROM Organization for 2 Intervals (continued)**

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
MISR for interval 1, when rom_access_inversion =0	MISR_GOLDEN[895:840]								
	MISR_GOLDEN[839:784]								
	MISR_GOLDEN[783:728]								
	MISR_GOLDEN[727:672]								
	MISR_GOLDEN[671:616]								
	MISR_GOLDEN[615:560]								
	MISR_GOLDEN[559:504]								
	MISR_GOLDEN[503:448]								
	MISR_GOLDEN[447:392]								
	MISR_GOLDEN[391:336]								
	MISR_GOLDEN[335:280]								
	MISR_GOLDEN[279:224]								
	MISR_GOLDEN[223:168]								
	MISR_GOLDEN[167:112]								
	MISR_GOLDEN[111:56]								
	MISR_GOLDEN[55:0]								
LP_MISR for interval 1, when rom_access_inversion =0	LP_MISR_GOLDEN[895:840]								
	LP_MISR_GOLDEN[839:784]								
	LP_MISR_GOLDEN[783:728]								
	LP_MISR_GOLDEN[727:672]								
	LP_MISR_GOLDEN[671:616]								
	LP_MISR_GOLDEN[615:560]								
	LP_MISR_GOLDEN[559:504]								
	LP_MISR_GOLDEN[503:448]								
	LP_MISR_GOLDEN[447:392]								
	LP_MISR_GOLDEN[391:336]								
	LP_MISR_GOLDEN[335:280]								
	LP_MISR_GOLDEN[279:224]								
	LP_MISR_GOLDEN[223:168]								
	LP_MISR_GOLDEN[167:112]								
	LP_MISR_GOLDEN[111:56]								
	LP_MISR_GOLDEN[55:0]								
Patterns for interval 1	P1_SD8[6:0 ]	P1_SD7[6:0 ]	P1_SD6[6 :0]	....	....	...	P1_SD1[ 6:0]		
		...	...	...	...	...	...	P1_SD9[ 6:0]	
		...	...	...	...	...	...	...	...

**Table 20-218. ROM Organization for 2 Intervals (continued)**

COMMENTS	55:40	41:32	31:16	15:8	7:4	3	2	1	0
LP_MISR for interval 1, when rom_access_inversion =1	LP_INV_MISR_GOLDEN[55:0]								
	LP_INV_MISR_GOLDEN[111:56]								
	LP_INV_MISR_GOLDEN[167:112]								
	LP_INV_MISR_GOLDEN[223:168]								
	LP_INV_MISR_GOLDEN[279:224]								
	LP_INV_MISR_GOLDEN[335:280]								
	LP_INV_MISR_GOLDEN[391:336]								
	LP_INV_MISR_GOLDEN[447:392]								
	LP_INV_MISR_GOLDEN[503:448]								
	LP_INV_MISR_GOLDEN[559:504]								
	LP_INV_MISR_GOLDEN[615:560]								
	LP_INV_MISR_GOLDEN[671:616]								
	LP_INV_MISR_GOLDEN[727:672]								
	LP_INV_MISR_GOLDEN[783:728]								
	LP_INV_MISR_GOLDEN[839:784]								
	LP_INV_MISR_GOLDEN[895:840]								
MISR for interval 1, when rom_access_inversion =1	INV_MISR_GOLDEN[55:0]								
	INV_MISR_GOLDEN[111:56]								
	INV_MISR_GOLDEN[167:112]								
	INV_MISR_GOLDEN[223:168]								
	INV_MISR_GOLDEN[279:224]								
	INV_MISR_GOLDEN[335:280]								
	INV_MISR_GOLDEN[391:336]								
	INV_MISR_GOLDEN[447:392]								
	INV_MISR_GOLDEN[503:448]								
	INV_MISR_GOLDEN[559:504]								
	INV_MISR_GOLDEN[615:560]								
	INV_MISR_GOLDEN[671:616]								
	INV_MISR_GOLDEN[727:672]								
	INV_MISR_GOLDEN[783:728]								
	INV_MISR_GOLDEN[839:784]								
	INV_MISR_GOLDEN[895:840]								
CFG for interval 1, when rom_access_inversion =1 (same as when_rom_access_inversion =0)	Reserved	pattern_count[9:0]	Reserved	Reserved	Reserved	Seg_ID[1]	Seg_ID[0]	FT	TR_T

The ROM contains the data to be processed by STC for the self-test run. This includes the control fields such as Segment ID, Pattern Count, and Golden MISR value for the STC, and the pattern scan data for the OPMISR controller.

The ROM space is divided into chunks, with each chunk containing the data corresponding to one OPMISR interval. The size required for an interval varies depending on the number patterns packed into the interval and the length of internal scan chains required.

Because each interval requires 64 rows of ROM for storing control and Golden MISR values, minimizing the number of intervals by packing more patterns into each interval provides the best ROM size. This works best if

the self-test must be run only as a part of the boot-up sequence. However, if the self-test is performed during application IDLE time, the number of patterns that can be packed into each interval will be dictated by the IDLE time available for the self-test, because an interval is the smallest granularity of a self-test run.

Details of the ROM image micro-code fields are given in the following sections.

#### 20.7.4.7.1 TR\_T: Transition Delay Methodology Type

This specifies the transition delay methodology for the current transition delay interval.

0	Launch-on-System-Clock
---	------------------------

#### 20.7.4.7.2 FT: Fault Model for the BIST Run

This specifies the fault model for the current interval of the test.

0	Stuck-at
---	----------

#### 20.7.4.7.3 SEG\_ID[1:0]

This indicates which logical segment is selected for the associated interval during the self-test run.

SEG_SEL[1:0]	Segment Under Test
00	Segment 0
01	Segment 1
10	Segment 2
11	Segment 3

#### 20.7.4.7.4 Pattern Count ( patt\_count[9:0] )

This specifies the number of scan data patterns within a self-test interval. The pattern counts can vary from a minimum of 2 to a maximum of 1024.

patt_count[9:0]	Patterns per Interval
00_0000_0000	Not a valid interval [defaults to 2 patterns per interval]
00_0000_0001	2 patterns per interval
00_0000_0010	3 patterns per interval
...	...
11_1111_1110	1023 patterns per interval
11_1111_1111	1024 patterns per interval

#### 20.7.4.7.5 MISR\_GOLDEN[895:0]: Golden Signature Data Bits

This part of ROM contains the golden signature data of the current interval. This value is used to compare with the actual MISR value, when ST\_GCR1.ROM\_ACCESS\_INV=0 and ST\_GCR1.LP\_SCAN\_MODE=0, to generate the pass/fail information of the interval.

**20.7.4.7.6 LP\_MISR\_GOLDEN[895:0]: Low Power Mode Golden Signature Data Bits**

This part of ROM contains the LP golden signature data of the current interval. This value is used to compare with the actual MISR value, when STCGCR1.ROM\_ACCESS\_INV=0 and STCGCR1.LP\_SCAN\_MODE=1, to generate the pass/fail information of the interval.

---

**Note**

This mode is not supported on AM26xx devices.

---

**20.7.4.7.7 INV\_MISR\_GOLDEN[895:0]: Inverse Mode Golden Signature Data Bits**

This part of ROM contains the inverse mode golden signature data of the current interval. This value is used to compare with the actual MISR value, when STCGCR1.ROM\_ACCESS\_INV=1 and STCGCR1.LP\_SCAN\_MODE=0, to generate the pass/fail information of the interval.

---

**Note**

This mode is not supported on AM26xx devices.

---

**20.7.4.7.8 LP\_INV\_MISR\_GOLDEN[895:0]: Low Power Inverse Mode Golden Signature Data Bits**

This part of ROM contains the low-power inverse mode golden signature data of the current interval. This value is used to compare with the actual MISR value, when STCGCR1.ROM\_ACCESS\_INV=1 and STCGCR1.LP\_SCAN\_MODE=1, to generate the pass/fail information of the interval.

---

**Note**

This mode is not supported on AM26xx devices.

---

**20.7.4.7.9 Pn\_SDm[7:0] (n - no. of patterns, m - scan chain length): OP-MISR Scan Data**

This part of the ROM contains the scan data corresponding to each pattern. Each interval can have n number of scan patterns, as defined in the patt\_count field. The number of 7bits of scan data in a pattern is equal to the length of the scan chain formed inside the UUT.

## 20.7.5 STC Registers

**Table 20-219** lists the memory-mapped registers for the STC registers. These registers are applicable for the R5F\_STC, DSP\_STC, and HSM\_STC. All register offset addresses not listed in **Table 20-219** should be considered as reserved locations and the register contents should not be modified.

**Table 20-219. STC Registers**

Offset	Acronym	Register Name	Section
0h	STCGCR0	Self test Global control Reg0	<a href="#">Go</a>
4h	STCGCR1	Self test Global control Reg1	<a href="#">Go</a>
8h	STCTPR	Time out counter preload register	<a href="#">Go</a>
Ch	STC_CADDR	Current Address register for CORE1	<a href="#">Go</a>
10h	STCCICR	Current Interval count register	<a href="#">Go</a>
14h	STCGSTAT	Global Status Register	<a href="#">Go</a>
18h	STCFSTAT	Fail Status Register	<a href="#">Go</a>
1Ch	STCSCSCR	Signature compare Self Check Register	<a href="#">Go</a>
20h	STC_CADDR2	Current Address register for CORE2	<a href="#">Go</a>
24h	STC_CLKDIV	Clock Divider Register	<a href="#">Go</a>
28h	STC_SEGPLR	Segment 1st interval Preload Register	<a href="#">Go</a>
2Ch	SEG0_START_ADDR	ROM Start address for Segment0	<a href="#">Go</a>
30h	SEG1_START_ADDR	ROM Start address for Segment1	<a href="#">Go</a>
34h	SEG2_START_ADDR	ROM Start address for Segment2	<a href="#">Go</a>
38h	SEG3_START_ADDR	ROM Start address for Segment3	<a href="#">Go</a>
3Ch	CORE1_CURMISR_0	Holds the MISR signature for CORE1	<a href="#">Go</a>
40h	CORE1_CURMISR_1	Holds the MISR signature for CORE1	<a href="#">Go</a>
44h	CORE1_CURMISR_2	Holds the MISR signature for CORE1	<a href="#">Go</a>
48h	CORE1_CURMISR_3	Holds the MISR signature for CORE1	<a href="#">Go</a>
4Ch	CORE1_CURMISR_4	Holds the MISR signature for CORE1	<a href="#">Go</a>
50h	CORE1_CURMISR_5	Holds the MISR signature for CORE1	<a href="#">Go</a>
54h	CORE1_CURMISR_6	Holds the MISR signature for CORE1	<a href="#">Go</a>
58h	CORE1_CURMISR_7	Holds the MISR signature for CORE1	<a href="#">Go</a>
5Ch	CORE1_CURMISR_8	Holds the MISR signature for CORE1	<a href="#">Go</a>
60h	CORE1_CURMISR_9	Holds the MISR signature for CORE1	<a href="#">Go</a>
64h	CORE1_CURMISR_10	Holds the MISR signature for CORE1	<a href="#">Go</a>
68h	CORE1_CURMISR_11	Holds the MISR signature for CORE1	<a href="#">Go</a>
6Ch	CORE1_CURMISR_12	Holds the MISR signature for CORE1	<a href="#">Go</a>
70h	CORE1_CURMISR_13	Holds the MISR signature for CORE1	<a href="#">Go</a>
74h	CORE1_CURMISR_14	Holds the MISR signature for CORE1	<a href="#">Go</a>
78h	CORE1_CURMISR_15	Holds the MISR signature for CORE1	<a href="#">Go</a>
7Ch	CORE1_CURMISR_16	Holds the MISR signature for CORE1	<a href="#">Go</a>
80h	CORE1_CURMISR_17	Holds the MISR signature for CORE1	<a href="#">Go</a>
84h	CORE1_CURMISR_18	Holds the MISR signature for CORE1	<a href="#">Go</a>
88h	CORE1_CURMISR_19	Holds the MISR signature for CORE1	<a href="#">Go</a>
8Ch	CORE1_CURMISR_20	Holds the MISR signature for CORE1	<a href="#">Go</a>
90h	CORE1_CURMISR_21	Holds the MISR signature for CORE1	<a href="#">Go</a>
94h	CORE1_CURMISR_22	Holds the MISR signature for CORE1	<a href="#">Go</a>
98h	CORE1_CURMISR_23	Holds the MISR signature for CORE1	<a href="#">Go</a>
9Ch	CORE1_CURMISR_24	Holds the MISR signature for CORE1	<a href="#">Go</a>



**Table 20-219. STC Registers (continued)**

Offset	Acronym	Register Name	Section
A0h	CORE1_CURMISR_25	Holds the MISR signature for CORE1	<a href="#">Go</a>
A4h	CORE1_CURMISR_26	Holds the MISR signature for CORE1	<a href="#">Go</a>
A8h	CORE1_CURMISR_27	Holds the MISR signature for CORE1	<a href="#">Go</a>
ACH	CORE2_CURMISR_0	Holds the MISR signature for CORE2	<a href="#">Go</a>
B0h	CORE2_CURMISR_1	Holds the MISR signature for CORE2	<a href="#">Go</a>
B4h	CORE2_CURMISR_2	Holds the MISR signature for CORE2	<a href="#">Go</a>
B8h	CORE2_CURMISR_3	Holds the MISR signature for CORE2	<a href="#">Go</a>
BCh	CORE2_CURMISR_4	Holds the MISR signature for CORE2	<a href="#">Go</a>
C0h	CORE2_CURMISR_5	Holds the MISR signature for CORE2	<a href="#">Go</a>
C4h	CORE2_CURMISR_6	Holds the MISR signature for CORE2	<a href="#">Go</a>
C8h	CORE2_CURMISR_7	Holds the MISR signature for CORE2	<a href="#">Go</a>
CCh	CORE2_CURMISR_8	Holds the MISR signature for CORE2	<a href="#">Go</a>
D0h	CORE2_CURMISR_9	Holds the MISR signature for CORE2	<a href="#">Go</a>
D4h	CORE2_CURMISR_10	Holds the MISR signature for CORE2	<a href="#">Go</a>
D8h	CORE2_CURMISR_11	Holds the MISR signature for CORE2	<a href="#">Go</a>
DCh	CORE2_CURMISR_12	Holds the MISR signature for CORE2	<a href="#">Go</a>
E0h	CORE2_CURMISR_13	Holds the MISR signature for CORE2	<a href="#">Go</a>
E4h	CORE2_CURMISR_14	Holds the MISR signature for CORE2	<a href="#">Go</a>
E8h	CORE2_CURMISR_15	Holds the MISR signature for CORE2	<a href="#">Go</a>
ECh	CORE2_CURMISR_16	Holds the MISR signature for CORE2	<a href="#">Go</a>
F0h	CORE2_CURMISR_17	Holds the MISR signature for CORE2	<a href="#">Go</a>
F4h	CORE2_CURMISR_18	Holds the MISR signature for CORE2	<a href="#">Go</a>
F8h	CORE2_CURMISR_19	Holds the MISR signature for CORE2	<a href="#">Go</a>
FCh	CORE2_CURMISR_20	Holds the MISR signature for CORE2	<a href="#">Go</a>
100h	CORE2_CURMISR_21	Holds the MISR signature for CORE2	<a href="#">Go</a>
104h	CORE2_CURMISR_22	Holds the MISR signature for CORE2	<a href="#">Go</a>
108h	CORE2_CURMISR_23	Holds the MISR signature for CORE2	<a href="#">Go</a>
10Ch	CORE2_CURMISR_24	Holds the MISR signature for CORE2	<a href="#">Go</a>
110h	CORE2_CURMISR_25	Holds the MISR signature for CORE2	<a href="#">Go</a>
114h	CORE2_CURMISR_26	Holds the MISR signature for CORE2	<a href="#">Go</a>
118h	CORE2_CURMISR_27	Holds the MISR signature for CORE2	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 20-220](#) shows the codes that are used for access types in this section.

**Table 20-220. STC Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 20.7.5.1 STCGCR0 Register (Offset = 0h) [Reset = 00000000h]

STCGCR0 is shown in [Table 20-221](#).

Return to the [Summary Table](#).

Self test Global control Reg0. \*NOT BYTE ACCESSIBLE

**Table 20-221. STCGCR0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	INTCOUNT_B16	R/W	0h	Number of intervals of the self test run (RWP - Read, Privilege Mode Write only) Count of intervals that need to be covered for a specific selftest run. The selftest controller sends out "complete" indication once it runs all of the intervals programmed in this field. INTCOUNT_B 16= 0 is an invalid configuration for a selftest.
15-11	NU0	R	0h	Reserved bits
10-8	CAP_IDLE_CYCLE	R/W	0h	Idle cycles before and after capture clock (RWP - Read, Privilege Mode Write only) Idle Cycles before and after capture clock. This value is used to insert that many idle cycles in the Capture phase. Programmable idle cycles allow implementation flexibility on SCAN_EN signal at chip level based on the size of the UUT and timing requirements.
7-5	SCANEN_HIGH_CAP_IDLE_CYCLE	R/W	0h	Idle cycles before and after capture clock (RWP - Read, Privilege Mode Write only). *NOT BYTE ACCESSIBLE Idle Cycles between scan_en going high to func_clk_en generation and scan_en going high to misr_log_en generation. This value is used to insert that many idle cycles in the shift clock (scan_en going high to func_clk_en generation) and misr_log_clk (scan_en going high to misr_log_en generation) generation. Programmable idle cycles allow implementation flexibility on SCAN_EN signal at chip level based on the size of the UUT and timing requirements.
4-2	NU1	R	0h	Reserved bits
1-0	RS_CNT_B1	R/W	0h	Restart/Continue or preload (RWP - Read, Privilege Mode Write only) This bit specifies the selftest controller whether to continue the run from next interval onwards, restart from ROM address 0 or preload from a prescribed interval. This bit gets reset after the completion of selftest run. 00 = Continue NSTC run from previous interval 01 = Restart NSTC run from ROM address 0 1X = Start from segment number specified in STC_SEGPLR register

### 20.7.5.2 STCGCR1 Register (Offset = 4h) [Reset = 00000000h]

STCGCR1 is shown in [Table 20-222](#).

Return to the [Summary Table](#).

Self test Global control Reg1

**Table 20-222. STCGCR1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	NU2	R	0h	Reserved bits
11-8	SEG0_CORE_SEL	R/W	0h	Selects the Segment 0 CORE for self test (RWP - Read, Priviledge Mode Write only) Select the Segment 0 CORE for Self -Test 0001 = Select CORE for selftest Other = CORE not selected.
7	NU3	R	0h	Reserved bits
6	CODEC_SPREAD_MODE	R/W	0h	Codec Spread Mode control signal (RWP - Read, Priviledge Mode Write only) This bit is used to configure the codec in spread / X-OR mode. 1 = Spread mode 0 = XOR mode
5	LP_SCAN_MODE	R/W	0h	LP scan mode (RWP - Read, Priviledge Mode Write only) This bit is used to decide the scan configuration: 1 = Operates in Low Power Scan Mode. 0 = Operates in Normal Scan Mode.
4	ROM_ACCESS_INV	R/W	0h	Rom access inversion mode (RWP - Read, Priviledge Mode Write only) - NOT SUPPORTED
3-0	ST_ENA_B4	R/W	0h	Self test enable key (RWP - Read, Priviledge Mode Write only) 1010 = Self test run enabled All values other than 1010 = Self test run disabled

### 20.7.5.3 STCTPR Register (Offset = 8h) [Reset = 00000000h]

STCTPR is shown in [Table 20-223](#).

Return to the [Summary Table](#).

Time out counter preload register

**Table 20-223. STCTPR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	TO_PRELOAD	R/W	0h	<p>Self test time out preload (RWP - Read, Priviledge Mode Write only)</p> <p>This register contains the total number of STC clock cycles it will take before a self-test timeout error will be triggered after the initiation of the self-test run.</p> <p>This is a fail safe feature to avoid system hang-up situation on account of any run away self test issues.</p> <p>This register should be loaded with a meaningful count value for this feature to be effective.</p> <p>This register value (preload count value) gets loaded into the self test timeout down counter whenever a self test run is initiated (ST_ENA is enabled).</p> <p>and gets disabled on completion of a self test run.</p>

#### 20.7.5.4 STC\_CADDR Register (Offset = Ch) [Reset = 00000000h]

STC\_CADDR is shown in [Table 20-224](#).

Return to the [Summary Table](#).

Current Address register for CORE1

**Table 20-224. STC\_CADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Current ROM Address for CORE1 This register reflects the current ROM address (for micro code load) accessed during selftest for CORE1 in of case segment0 and all the remaining segmentsn where n = 1 to 3).

### 20.7.5.5 STCCICR Register (Offset = 10h) [Reset = 00000000h]

STCCICR is shown in [Table 20-225](#).

Return to the [Summary Table](#).

Current Interval count register

**Table 20-225. STCCICR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	CORE2_ICOUNT	R	0h	Specifies the last interval number for CORE2 This specifies the Last executed Interval number for CORE2 of Segment0 if self test is being executed for secondary core as well. This field is applicable only for Segment 0.
15-0	CORE1_ICOUNT	R	0h	Specifies the last interval number for CORE1 This specifies the Last executed Interval number of a self-test run.

### 20.7.5.6 STCGSTAT Register (Offset = 14h) [Reset = 00000000h]

STCGSTAT is shown in [Table 20-226](#).

Return to the [Summary Table](#).

Global Status Register

**Table 20-226. STCGSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-12	NU4	R	0h	Reserved bits
11-8	ST_ACTIVE	R	0h	Tells whether self test is currently active or not. 1010 = Self test is active Others = SelfTest is not active Once the self-test completes and ST_ENA_B 4 key is cleared, this field will reflect the inactive value.
7-2	NU5	R	0h	Reserved bits
1	TEST_FAIL	R	0h	Test_fail flag (RCP - Read, Clear on Writing in Priviledge Mode) 0 = Self test run has not failed 1 = SelfTest run has failed. Write Clear.
0	TEST_DONE	R	0h	Test_done_flag (RCP - Read, Clear on Writing in Priviledge Mode) 0 = Not completed 1 = SelfTest run Completed

### 20.7.5.7 STCFSTAT Register (Offset = 18h) [Reset = 00000000h]

STCFSTAT is shown in [Table 20-227](#).

Return to the [Summary Table](#).

Fail Status Register

**Table 20-227. STCFSTAT Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	NU6	R	0h	Reserved bits
4-3	FSEG_ID	R	0h	Failed Segment ID (RCP - Read, Clear on Writing in Privilege Mode) This field captures the Segment number for which any of the failures like TO_ER_B 1, CPU 1_FAIL_B 1 and CPU 2_FAIL_B 1 occur. 00 = Failure on Segment 0 01 = Failure on Segment 1 10 = Failure on Segment 2 11 = Failure on Segment 3
2	TO_ER_B1	R	0h	Tells whether self test failed because of time out error (RCP - Read, Clear on Writing in Privilege Mode) 0 = No time out error occurred 1 = SelfTest run failed due to a timeout error
1	CPU2_FAIL_B1	R	0h	Tells whether MISR mismatch happened in CORE 2 when in Segment 0 mode (RCP - Read, Clear on Writing in Privilege Mode) 0 = No MISR mismatch for CORE 2 1 = Self test run failed due to MISR mismatch for CORE 2
0	CPU1_FAIL_B1	R	0h	Tells whether MISR mismatch happened in CORE 1 (RCP - Read, Clear on Writing in Privilege Mode) Applicable to all segments. 0 = No MISR mismatch for CORE 1 1 = Self test run failed due to MISR mismatch for CORE 1



### 20.7.5.8 STCSCSCR Register (Offset = 1Ch) [Reset = 00000000h]

STCSCSCR is shown in [Table 20-228](#).

Return to the [Summary Table](#).

Signature compare Self Check Register

**Table 20-228. STCSCSCR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-5	NU7	R	0h	Reserved bits
4	FAULT_INS_B1	R/W	0h	Fault Insertion bit (RWP - Read, Priviledge Mode Write only) 0 = No fault insertion. 1 = Inserts fault in the logic unedr test which will make signature compare fail. This feature is used as diagnostic check of the STC IP.
3-0	SELF_CHECK_KEY_B4	R/W	0h	Signature compare logic self check key enable/disable (RWP - Read, Priviledge Mode Write only) 1010 = Signature compare logic Self Check is enabled All values other than 1010 = Signature compare logic Self Check is disabled

### 20.7.5.9 STC\_CADDR2 Register (Offset = 20h) [Reset = 00000000h]

STC\_CADDR2 is shown in [Table 20-229](#).

Return to the [Summary Table](#).

Current Address register for CORE2

**Table 20-229. STC\_CADDR2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	ADDR	R	0h	Current ROM Address for CORE2 This register reflects the current ROM address(for micro code load) accessed during selftest for CORE2 in of case segment0.

### 20.7.5.10 STC\_CLKDIV Register (Offset = 24h) [Reset = 00000000h]

STC\_CLKDIV is shown in [Table 20-230](#).

Return to the [Summary Table](#).

Clock Divider Register

**Table 20-230. STC\_CLKDIV Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-27	NU8	R	0h	Reserved bits
26-24	CLKDIV0	R/W	0h	Clock division for Seg0 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 0
23-19	NU9	R	0h	Reserved bits
18-16	CLKDIV1	R/W	0h	Clock division for Seg1 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 1
15-11	NU10	R	0h	Reserved bits
10-8	CLKDIV2	R/W	0h	Clock division for Seg2 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 2
7-3	NU11	R	0h	Reserved bits
2-0	CLKDIV3	R/W	0h	Clock division for Seg3 (RWP - Read, Priviledge Mode Write only) *NOT SUPPORTED X = Division ratio is X+1 for Segment 3

### 20.7.5.11 STC\_SEGPLR Register (Offset = 28h) [Reset = 00000000h]

STC\_SEGPLR is shown in [Table 20-231](#).

Return to the [Summary Table](#).

Segment 1st interval Preload Register

**Table 20-231. STC\_SEGPLR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-2	NU12	R	0h	Reserved bits
1-0	SEGID_PLOAD	R/W	0h	<p>Segment number for which preload is to be started (RWP - Read, Priviledge Mode Write only) This specifies the segment for which the address of its First interval will be pre-loaded into the NSTC ROM address counter.</p> <p>The</p> <p>1st address of each segment are defined in SEGx_START_ADDR register. The address of the</p> <p>1st interval of the selected segment is loaded into the NSTC ROM address counter when the RS_CNT_B</p> <p>1 bits of STC_GCR</p> <p>0 are set to</p> <p>1X</p> <p>00 = Preload the address of the</p> <p>1st interval of segment</p> <p>0.</p> <p>01 = Preload the address of the</p> <p>1st interval of segment</p> <p>1.</p> <p>10 = Preload the address of the</p> <p>1st interval of segment</p> <p>2.</p> <p>11 = Preload the address of the</p> <p>1st interval of segment</p> <p>3.</p>

### 20.7.5.12 SEG0\_START\_ADDR Register (Offset = 2Ch) [Reset = 00000000h]

SEG0\_START\_ADDR is shown in [Table 20-232](#).

Return to the [Summary Table](#).

ROM Start address for Segment0

**Table 20-232. SEG0\_START\_ADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU13	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 0 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) "PRELOAD" option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register. Valid number of bits depends on RTL paramerter ADDR

### 20.7.5.13 SEG1\_START\_ADDR Register (Offset = 30h) [Reset = 00000000h]

SEG1\_START\_ADDR is shown in [Table 20-233](#).

Return to the [Summary Table](#).

ROM Start address for Segment1

**Table 20-233. SEG1\_START\_ADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU14	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 1 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) "PRELOAD" option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register. Valid number of bits depends on RTL parameter ADDR. This register is present only when RTL parameter NUM_SEG = 1.

#### 20.7.5.14 SEG2\_START\_ADDR Register (Offset = 34h) [Reset = 00000000h]

SEG2\_START\_ADDR is shown in [Table 20-234](#).

Return to the [Summary Table](#).

ROM Start address for Segment2

**Table 20-234. SEG2\_START\_ADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU15	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 2 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) "PRELOAD" option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register. Valid number of bits depends on RTL parameter ADDR. This register is present only when RTL parameter NUM_SEG = 2.

### 20.7.5.15 SEG3\_START\_ADDR Register (Offset = 38h) [Reset = 00000000h]

SEG3\_START\_ADDR is shown in [Table 20-235](#).

Return to the [Summary Table](#).

ROM Start address for Segment3

**Table 20-235. SEG3\_START\_ADDR Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU16	R	0h	Reserved bits
19-0	SEG_START_ADDR	R/W	0h	Segment 3 Start Address (RWP - Read, Priviledge Mode Write only) This register holds the ROM address for the start of first interval of the segment. When STC_GCR0.RS_CNT_B1 field is set to (1x) "PRELOAD" option, this register is used to determine the ROM start address for the Segment selected in ST_SEGPLR register. Valid number of bits depends on RTL parameter ADDR. This register is present only when RTL parameter NUM_SEG = 3.



### 20.7.5.16 CORE1\_CURMISR\_0 Register (Offset = 3Ch) [Reset = 00000000h]

CORE1\_CURMISR\_0 is shown in [Table 20-236](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-236. CORE1\_CURMISR\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR0	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.17 CORE1\_CURMISR\_1 Register (Offset = 40h) [Reset = 00000000h]

CORE1\_CURMISR\_1 is shown in [Table 20-237](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-237. CORE1\_CURMISR\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR1	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.18 CORE1\_CURMISR\_2 Register (Offset = 44h) [Reset = 00000000h]

CORE1\_CURMISR\_2 is shown in [Table 20-238](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-238. CORE1\_CURMISR\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR2	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.19 CORE1\_CURMISR\_3 Register (Offset = 48h) [Reset = 00000000h]

CORE1\_CURMISR\_3 is shown in [Table 20-239](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-239. CORE1\_CURMISR\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR3	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.20 CORE1\_CURMISR\_4 Register (Offset = 4Ch) [Reset = 00000000h]

CORE1\_CURMISR\_4 is shown in [Table 20-240](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-240. CORE1\_CURMISR\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR4	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.21 CORE1\_CURMISR\_5 Register (Offset = 50h) [Reset = 00000000h]

CORE1\_CURMISR\_5 is shown in [Table 20-241](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-241. CORE1\_CURMISR\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR5	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.22 CORE1\_CURMISR\_6 Register (Offset = 54h) [Reset = 00000000h]

CORE1\_CURMISR\_6 is shown in [Table 20-242](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-242. CORE1\_CURMISR\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR6	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.23 CORE1\_CURMISR\_7 Register (Offset = 58h) [Reset = 00000000h]

CORE1\_CURMISR\_7 is shown in [Table 20-243](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-243. CORE1\_CURMISR\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR7	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>



#### 20.7.5.24 CORE1\_CURMISR\_8 Register (Offset = 5Ch) [Reset = 00000000h]

CORE1\_CURMISR\_8 is shown in [Table 20-244](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-244. CORE1\_CURMISR\_8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR8	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.25 CORE1\_CURMISR\_9 Register (Offset = 60h) [Reset = 00000000h]

CORE1\_CURMISR\_9 is shown in [Table 20-245](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-245. CORE1\_CURMISR\_9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR9	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.26 CORE1\_CURMISR\_10 Register (Offset = 64h) [Reset = 00000000h]

CORE1\_CURMISR\_10 is shown in [Table 20-246](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-246. CORE1\_CURMISR\_10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR10	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.27 CORE1\_CURMISR\_11 Register (Offset = 68h) [Reset = 00000000h]

CORE1\_CURMISR\_11 is shown in [Table 20-247](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-247. CORE1\_CURMISR\_11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR11	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.28 CORE1\_CURMISR\_12 Register (Offset = 6Ch) [Reset = 00000000h]

CORE1\_CURMISR\_12 is shown in [Table 20-248](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-248. CORE1\_CURMISR\_12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR12	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.29 CORE1\_CURMISR\_13 Register (Offset = 70h) [Reset = 00000000h]

CORE1\_CURMISR\_13 is shown in [Table 20-249](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-249. CORE1\_CURMISR\_13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR13	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.30 CORE1\_CURMISR\_14 Register (Offset = 74h) [Reset = 00000000h]

CORE1\_CURMISR\_14 is shown in [Table 20-250](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-250. CORE1\_CURMISR\_14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR14	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.31 CORE1\_CURMISR\_15 Register (Offset = 78h) [Reset = 00000000h]

CORE1\_CURMISR\_15 is shown in [Table 20-251](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-251. CORE1\_CURMISR\_15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR15	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>



### 20.7.5.32 CORE1\_CURMISR\_16 Register (Offset = 7Ch) [Reset = 00000000h]

CORE1\_CURMISR\_16 is shown in [Table 20-252](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-252. CORE1\_CURMISR\_16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR16	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.33 CORE1\_CURMISR\_17 Register (Offset = 80h) [Reset = 00000000h]

CORE1\_CURMISR\_17 is shown in [Table 20-253](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-253. CORE1\_CURMISR\_17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR17	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.34 CORE1\_CURMISR\_18 Register (Offset = 84h) [Reset = 00000000h]

CORE1\_CURMISR\_18 is shown in [Table 20-254](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-254. CORE1\_CURMISR\_18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR18	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.35 CORE1\_CURMISR\_19 Register (Offset = 88h) [Reset = 00000000h]

CORE1\_CURMISR\_19 is shown in [Table 20-255](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-255. CORE1\_CURMISR\_19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR19	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.36 CORE1\_CURMISR\_20 Register (Offset = 8Ch) [Reset = 00000000h]

CORE1\_CURMISR\_20 is shown in [Table 20-256](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-256. CORE1\_CURMISR\_20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR20	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.37 CORE1\_CURMISR\_21 Register (Offset = 90h) [Reset = 00000000h]

CORE1\_CURMISR\_21 is shown in [Table 20-257](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-257. CORE1\_CURMISR\_21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR21	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.38 CORE1\_CURMISR\_22 Register (Offset = 94h) [Reset = 00000000h]

CORE1\_CURMISR\_22 is shown in [Table 20-258](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-258. CORE1\_CURMISR\_22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR22	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.39 CORE1\_CURMISR\_23 Register (Offset = 98h) [Reset = 00000000h]

CORE1\_CURMISR\_23 is shown in [Table 20-259](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-259. CORE1\_CURMISR\_23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR23	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>



#### 20.7.5.40 CORE1\_CURMISR\_24 Register (Offset = 9Ch) [Reset = 00000000h]

CORE1\_CURMISR\_24 is shown in [Table 20-260](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-260. CORE1\_CURMISR\_24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR24	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

#### 20.7.5.41 CORE1\_CURMISR\_25 Register (Offset = A0h) [Reset = 00000000h]

CORE1\_CURMISR\_25 is shown in [Table 20-261](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-261. CORE1\_CURMISR\_25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR25	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

#### 20.7.5.42 CORE1\_CURMISR\_26 Register (Offset = A4h) [Reset = 00000000h]

CORE1\_CURMISR\_26 is shown in [Table 20-262](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-262. CORE1\_CURMISR\_26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR26	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.43 CORE1\_CURMISR\_27 Register (Offset = A8h) [Reset = 00000000h]

CORE1\_CURMISR\_27 is shown in [Table 20-263](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE1

**Table 20-263. CORE1\_CURMISR\_27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C1MISR27	R	0h	<p>MISR Signature for CORE1 This register contains the MISR data of the current interval for CORE1 in the case of segment0 and the remaining Segments 1 to 3.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

#### 20.7.5.44 CORE2\_CURMISR\_0 Register (Offset = ACh) [Reset = 00000000h]

CORE2\_CURMISR\_0 is shown in [Table 20-264](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-264. CORE2\_CURMISR\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR0	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.45 CORE2\_CURMISR\_1 Register (Offset = B0h) [Reset = 00000000h]

CORE2\_CURMISR\_1 is shown in [Table 20-265](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-265. CORE2\_CURMISR\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR1	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.46 CORE2\_CURMISR\_2 Register (Offset = B4h) [Reset = 00000000h]

CORE2\_CURMISR\_2 is shown in [Table 20-266](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-266. CORE2\_CURMISR\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR2	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.47 CORE2\_CURMISR\_3 Register (Offset = B8h) [Reset = 00000000h]

CORE2\_CURMISR\_3 is shown in [Table 20-267](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-267. CORE2\_CURMISR\_3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR3	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>



#### 20.7.5.48 CORE2\_CURMISR\_4 Register (Offset = BCh) [Reset = 00000000h]

CORE2\_CURMISR\_4 is shown in [Table 20-268](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-268. CORE2\_CURMISR\_4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR4	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

#### 20.7.5.49 CORE2\_CURMISR\_5 Register (Offset = C0h) [Reset = 00000000h]

CORE2\_CURMISR\_5 is shown in [Table 20-269](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-269. CORE2\_CURMISR\_5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR5	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.50 CORE2\_CURMISR\_6 Register (Offset = C4h) [Reset = 00000000h]

CORE2\_CURMISR\_6 is shown in [Table 20-270](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-270. CORE2\_CURMISR\_6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR6	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.51 CORE2\_CURMISR\_7 Register (Offset = C8h) [Reset = 00000000h]

CORE2\_CURMISR\_7 is shown in [Table 20-271](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-271. CORE2\_CURMISR\_7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR7	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.52 CORE2\_CURMISR\_8 Register (Offset = CCh) [Reset = 00000000h]

CORE2\_CURMISR\_8 is shown in [Table 20-272](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-272. CORE2\_CURMISR\_8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR8	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.53 CORE2\_CURMISR\_9 Register (Offset = D0h) [Reset = 00000000h]

CORE2\_CURMISR\_9 is shown in [Table 20-273](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-273. CORE2\_CURMISR\_9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR9	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.54 CORE2\_CURMISR\_10 Register (Offset = D4h) [Reset = 00000000h]

CORE2\_CURMISR\_10 is shown in [Table 20-274](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-274. CORE2\_CURMISR\_10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR10	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.55 CORE2\_CURMISR\_11 Register (Offset = D8h) [Reset = 00000000h]

CORE2\_CURMISR\_11 is shown in [Table 20-275](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-275. CORE2\_CURMISR\_11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR11	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>



### 20.7.5.56 CORE2\_CURMISR\_12 Register (Offset = DCh) [Reset = 00000000h]

CORE2\_CURMISR\_12 is shown in [Table 20-276](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-276. CORE2\_CURMISR\_12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR12	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.57 CORE2\_CURMISR\_13 Register (Offset = E0h) [Reset = 00000000h]

CORE2\_CURMISR\_13 is shown in [Table 20-277](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-277. CORE2\_CURMISR\_13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR13	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.58 CORE2\_CURMISR\_14 Register (Offset = E4h) [Reset = 00000000h]

CORE2\_CURMISR\_14 is shown in [Table 20-278](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-278. CORE2\_CURMISR\_14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR14	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.59 CORE2\_CURMISR\_15 Register (Offset = E8h) [Reset = 00000000h]

CORE2\_CURMISR\_15 is shown in [Table 20-279](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-279. CORE2\_CURMISR\_15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR15	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.60 CORE2\_CURMISR\_16 Register (Offset = ECh) [Reset = 00000000h]

CORE2\_CURMISR\_16 is shown in [Table 20-280](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-280. CORE2\_CURMISR\_16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR16	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.61 CORE2\_CURMISR\_17 Register (Offset = F0h) [Reset = 00000000h]

CORE2\_CURMISR\_17 is shown in [Table 20-281](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-281. CORE2\_CURMISR\_17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR17	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.62 CORE2\_CURMISR\_18 Register (Offset = F4h) [Reset = 00000000h]

CORE2\_CURMISR\_18 is shown in [Table 20-282](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-282. CORE2\_CURMISR\_18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR18	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.63 CORE2\_CURMISR\_19 Register (Offset = F8h) [Reset = 00000000h]

CORE2\_CURMISR\_19 is shown in [Table 20-283](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-283. CORE2\_CURMISR\_19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR19	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>



### 20.7.5.64 CORE2\_CURMISR\_20 Register (Offset = FCh) [Reset = 00000000h]

CORE2\_CURMISR\_20 is shown in [Table 20-284](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-284. CORE2\_CURMISR\_20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR20	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.65 CORE2\_CURMISR\_21 Register (Offset = 100h) [Reset = 00000000h]

CORE2\_CURMISR\_21 is shown in [Table 20-285](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-285. CORE2\_CURMISR\_21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR21	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.66 CORE2\_CURMISR\_22 Register (Offset = 104h) [Reset = 00000000h]

CORE2\_CURMISR\_22 is shown in [Table 20-286](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-286. CORE2\_CURMISR\_22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR22	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.67 CORE2\_CURMISR\_23 Register (Offset = 108h) [Reset = 00000000h]

CORE2\_CURMISR\_23 is shown in [Table 20-287](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-287. CORE2\_CURMISR\_23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR23	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.68 CORE2\_CURMISR\_24 Register (Offset = 10Ch) [Reset = 00000000h]

CORE2\_CURMISR\_24 is shown in [Table 20-288](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-288. CORE2\_CURMISR\_24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR24	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.69 CORE2\_CURMISR\_25 Register (Offset = 110h) [Reset = 00000000h]

CORE2\_CURMISR\_25 is shown in [Table 20-289](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-289. CORE2\_CURMISR\_25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR25	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.70 CORE2\_CURMISR\_26 Register (Offset = 114h) [Reset = 00000000h]

CORE2\_CURMISR\_26 is shown in [Table 20-290](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-290. CORE2\_CURMISR\_26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR26	R	0h	<p>MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval.</p> <p>This is applicable to Segment 0 alone.</p> <p>This value will be compared with the GOLDEN MISR value copied from ROM.</p> <p>This register gets reset to its default value with Power on or system reset assertion.</p> <p>The MISR values should be read only after the Self Test is completed.</p>

### 20.7.5.71 CORE2\_CURMISR\_27 Register (Offset = 118h) [Reset = 00000000h]

CORE2\_CURMISR\_27 is shown in [Table 20-291](#).

Return to the [Summary Table](#).

Holds the MISR signature for CORE2

**Table 20-291. CORE2\_CURMISR\_27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	C2MISR27	R	0h	MISR Signature for CORE2 This register contains the MISR data from the CORE2 for the current interval. This is applicable to Segment 0 alone. This value will be compared with the GOLDEN MISR value copied from ROM. This register gets reset to its default value with Power on or system reset assertion. The MISR values should be read only after the Self Test is completed.

## 20.8 Programmable Built-In Self-Test (PBIST)

This section describes the programmable built-in self-test (PBIST) controller module used for testing the on-chip memories.

### 20.8.1 Overview

The PBIST (Programmable Built-In Self-Test) controller architecture provides a run-time-programmable memory BIST engine for varying levels of coverage across many embedded memory instances.

Name	Frame Address (Hex) Start	Frame Address (Hex) End	Size	Description
TOP_PBIST	0x02F7_9400	0x02F7_95CC	464Bytes	PBIST module configuration registers. This IP covers memory test for all memories excluding DSP L1P, L1D and associated TAG memories
DSS_DSP_PBIST	0x06F7_9000	0x06F7_91CC	464Bytes	PBIST module configuration registers. This IP covers memory test for all memories excluding DSP L1P, L1D and associated TAG memories

#### 20.8.1.1 PBIST vs. Application Software-Based Testing

The PBIST architecture consists of a small coprocessor with a dedicated instruction set targeted specifically toward testing memories. This coprocessor executes test routines stored in the PBIST ROM and runs them on multiple on-chip memory instances. The on-chip memory configuration information is also stored in the PBIST ROM.

The PBIST Controller architecture offers significant advantages over tests running on the main processor (application software-based testing):

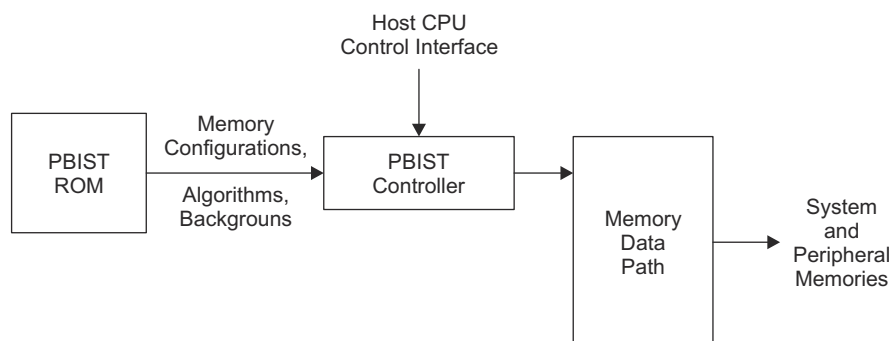
- Embedded CPUs have a long access path to memories outside the tightly-couple memory sub-system, while the PBIST controller has a dedicated path to the memories specifically for the self-test
- Embedded CPUs are designed for their targeted use and are often not easily programmed for memory test algorithms.
- The memory test algorithm code on embedded CPUs is typically significantly larger than that needed for PBIST.



- The embedded CPU is significantly larger than the PBIST controller.

### 20.8.1.2 PBIST Block Diagram

Figure 20-201 illustrates the basic PBIST blocks and its wrapper logic for the device.



**Figure 20-201. PBIST Block Diagram**

### 20.8.1.2.1 On-chip ROM

The on-chip ROM contains the information regarding the algorithms and memories to be tested.

### 20.8.1.2.2 Host Processor Interface to the PBIST Controller Registers

The CPU can select the algorithm and RAM groups for the memories' self-test from the onchip ROM based on the application requirements. Once the self-test has executed, the CPU can query the PBIST controller registers to identify any memories that failed the self-test and to then take appropriate next steps as required by the application's author.

### 20.8.1.2.3 Memory Data Path

This is the read and write data path logic between different system and peripheral memories tightly coupled to the PBIST memory interface. The PBIST controller executes each selected algorithm on each valid memory group sequentially until all the algorithms are executed.

#### Note

NOTE: Not all algorithms are designed to run on all RAM groups. If an algorithm is selected to run on an incompatible memory, this will result in a failure. Refer to and for RAM grouping and algorithm information.

## 20.8.2 RAM/ROM Grouping and Algorithm

### 20.8.2.1 RAM Algorithm: March13N

This section provides a brief description for some of the test algorithms used for memory self-test.

•

#### • March13N:

– March13N is the baseline test algorithm for SRAM testing. It provides the highest overall coverage.

The other algorithms provide additional coverage of otherwise missed boundary conditions of the SRAM operation.

– The concept behind the general march algorithm is to indicate:

- The bits around the bit cell do not affect the bit cell.
- The bit cell can be written and read as both a 1 and a 0.

– The basic operation of the march is to initialize the array to a know pattern, then march a different pattern through the memory.

– Type of faults detected by this algorithm:

- Address decoder faults
- Stuck-At faults
- Coupled faults
- State coupling faults
- Parametric faults
- Write recovery faults
- Read/write logic faults

### 20.8.2.2 Read/write logic faults

The triple read reads the array, all the way through, three times while summing the reads to compare the sums for all three read formats. The algorithm checks if there is enough margin in both the erasure and programming to operate at full speed with the CPU. This can be addressed with the XOR Read (Memory Contents XOR Memory Address). An error in the XOR Read indicates that the interaction between adjacent bit cells, being a different polarity, may be causing speed issues when the CPU exercises worstcase instruction sequencing. Each

read can be performed on any memory block, and an associated checksum is calculated to determine PASS or FAIL.

Type of faults detected by this algorithm:

- Address decoder faults
- Stuck-At faults
- Coupled faults
- State coupling faults
- Parametric faults
- Read logic faults

#### 20.8.2.3

---

#### Note

March13N is the most recommended algorithm for the memory self-test

---



## 21.1 GPADC Overview

The AWR2544 device implements a GPADC module on MainSS for safety monitoring the inputs, such as the temp sensor, voltage regulators, and so forth; both external and internal to the device. Features supported by the ADC module include:

- 10-bit ENOB 625Ksps ADC
- Full-scale range of GPADC input between 0 V and 1.8 V
- external I/Os and internal components -temperature sensors (3), supplies, bias reference, LDO output mapped to GPADC input
- Total capacitance, including parasitic and sampling cap, for each of GPADC pins be less than 14pf
- Single conversion mode
- Event pin and software command mechanism to trigger the conversion
  - Event signal from either of the GPIO (provides options with pinmuxing) as well as from RTI timer module
- Data RAM to store the conversion results (1K results)
- Storage of min, max, and sum of the samples captured per channel
- Mechanism to skip initial M samples and program the number of samples N to be collected every channel
- DMA trigger when conversion is completed or data RAM threshold is achieved
- Self-test logic support for input channel failure detection (open / short to power / short to ground)
- Calibration logic using BGAP, for offset error correction
- Separate operating supply and ground pins on package
- Capability to have an external reference voltage for GPADC

---

### Note

1. It is recommended to gate of the MSS\_GPADC\_CLK when not in operation. This gating will reduce the standard-cell activity around the transmitter and reduces the chance of interference.
  2. Write 3b'111 to MSS\_RCM:MSS\_GPADC\_CLK\_GATE:MSS\_GPADC\_CLK\_GATE\_GATED gates the MSS\_GPADC\_CLK.
-

## 21.2 Functional Description Modes of Operation

### 21.2.1 IFM Mode (Inter Frame Monitoring Mode)

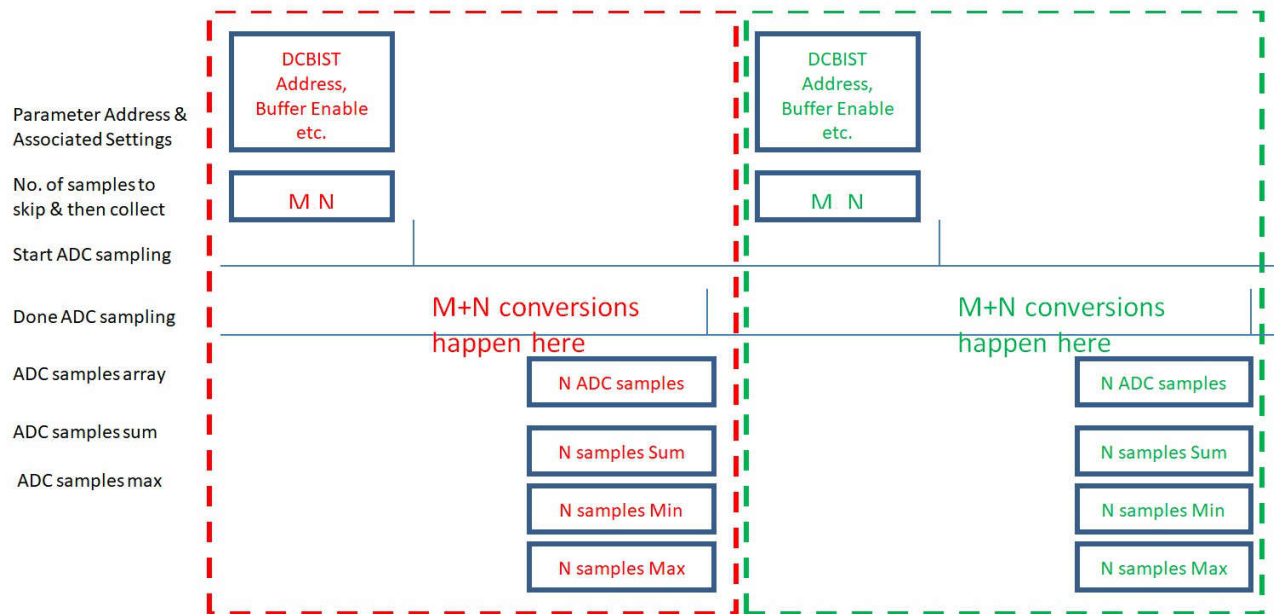


Figure 21-1. IFM Mode

Firmware writes the following register bit fields:

- Param\_val\_ifm
- Config\_value\_ifm
- Skip\_samples\_ifm(M)
- Collect\_samples\_ifm(N)
- Config settings in Analog
- ADC Reset =0
- ADC Enable =1
- ADC Start =1
- Firmware issues GPADC Trigger (self clearing bit)

Hardware skips M samples, collects N samples. Hardware outputs the sum, min, max of those samples in a register. Hardware outputs the actual array of samples in Data RAM (debug, if enabled). Each entry in the data RAM (256 deep) contains 4 ADC data samples, i.e. 1K ADC samples can be stored in the data RAM. Hardware issues a done interrupt.

## 21.3 Interrupts

The following interrupt is generated by GPADC:

- MSS\_GPADC\_IFM\_DONE

## 21.4 GPADC Trigger Sources

### 21.4.1 IFM Mode

Writing 1'b1 to MSS\_GPADC\_REG::REG1:: GPADC\_TRIGGER generates a trigger for IFM Mode to start.

## 21.5 MSS GPADC Register Specification

### 21.5.1 MSS\_GPADC\_REG Registers

Table 21-1 lists the MSS\_GPADC\_REG registers. All register offset addresses not listed in Table 21-1 should be considered as reserved locations and the register contents should not be modified.

**Table 21-1. MSS\_GPADC\_REG Registers**

Offset	Acronym	Register Name	Section
0h	REG0	GPADC_ENABLE_MODE	<a href="#">Section 21.5.1.1</a>
4h	REG1	GPADC_TRIGGER_IFM	<a href="#">Section 21.5.1.2</a>
8h	REG2	CFG_IFM	<a href="#">Section 21.5.1.3</a>
Ch	REG3	PARAM_COLLECT_SKIP_IFM	<a href="#">Section 21.5.1.4</a>
10h	REG4	BASE_ADDR_CP0_1_2_3	<a href="#">Section 21.5.1.5</a>
14h	REG5	BASE_ADDR_CP4_5_6_7	<a href="#">Section 21.5.1.6</a>
18h	REG6	BASE_ADDR_CP8_9_10_11	<a href="#">Section 21.5.1.7</a>
1Ch	REG7	BASE_ADDR_CP12_13_14_15	<a href="#">Section 21.5.1.8</a>
20h	REG8	GAPADC_CLK_CTRL	<a href="#">Section 21.5.1.9</a>
24h	REG9	param_not_used_tx_ena1_off	<a href="#">Section 21.5.1.10</a>
28h	REG10	param_not_used_tx_ena2_off	<a href="#">Section 21.5.1.11</a>
2Ch	REG11	param_not_used_tx_ena3_off	<a href="#">Section 21.5.1.12</a>
30h	REG12	SPARE1_WR	<a href="#">Section 21.5.1.13</a>
34h	REG13	SPARE2_WR	<a href="#">Section 21.5.1.14</a>
38h	REG14	SUM_IFM	<a href="#">Section 21.5.1.15</a>
3Ch	REG15	MIN_MAX_IFM	<a href="#">Section 21.5.1.16</a>
40h	REG16	GPADC_SAMPLES_FRAME	<a href="#">Section 21.5.1.17</a>
44h	REG17		<a href="#">Section 21.5.1.18</a>
48h	REG18		<a href="#">Section 21.5.1.19</a>
4Ch	REG19		<a href="#">Section 21.5.1.20</a>
50h	REG20		<a href="#">Section 21.5.1.21</a>
54h	REG21		<a href="#">Section 21.5.1.22</a>
58h	REG22		<a href="#">Section 21.5.1.23</a>

### 21.5.1.1 REG0 Register (Offset = 0h) [reset = 0h]

REG0 is shown in [Figure 21-2](#) and described in [Table 21-2](#).

Return to the [Table 21-1](#).

gpadc modes and enable

**Figure 21-2. REG0 Register**

31	30	29	28	27	26	25	24
NU3							
R-0h							
23	22	21	20	19	18	17	16
NU3						GPADC_DEBUG_MODE_ENABLE	
R-0h						R/W-0h	
15	14	13	12	11	10	9	8
NU2				GPADC2ADCBUF_PATH_EN			GPADC_FSM_CLK_ENABLE
R-0h				R/W-0h			R/W-0h
7	6	5	4	3	2	1	0
NU1						DCBIST_MODE	
R-0h						R/W-0h	

**Table 21-2. REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-17	NU3	R	0h	TI reserved
16	GPADC_DEBUG_MODE_ENABLE	R/W	0h	1:GPADC raw samples will be collected in the Output RAM in IFM mode
15-12	NU2	R	0h	TI reserved
11-9	GPADC2ADCBUF_PATH_EN	R/W	0h	TI reserved
8	GPADC_FSM_CLK_ENABLE	R/W	0h	Enable the clock to gpadc fsm
7-2	NU1	R	0h	TI reserved
1-0	DCBIST_MODE	R/W	0h	0:Disable,1:IFM Mode enable ,2:CTM mode enable

### 21.5.1.2 REG1 Register (Offset = 4h) [reset = 0h]

REG1 is shown in [Figure 21-3](#) and described in [Table 21-3](#).

Return to the [Table 21-1](#).

gpadc start trigger for Inter frame mode

**Figure 21-3. REG1 Register**

31	30	29	28	27	26	25	24
NU4							GPADC_START_BYP_VAL
R-0h							R/W-0h
23	22	21	20	19	18	17	16
NU3							GPADC_FSM_BYPASS
R-0h							R/W-0h
15	14	13	12	11	10	9	8
NU2							GPADC_INIT
R-0h							0h
7	6	5	4	3	2	1	0
NU1							GPADC_TRIGGER
R-0h							0h

**Table 21-3. REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-25	NU4	R	0h	TI reserved
24	GPADC_START_BYP_VAL	R/W	0h	
23-17	NU3	R	0h	TI reserved
16	GPADC_FSM_BYPASS	R/W	0h	1:Bypass gpadc control .When bypassed start = gpadc_start_byp_val config_val = config_value_ifm param_val = param_val_ifm
15-9	NU2	R	0h	TI reserved
8	GPADC_INIT		0h	Resets the FSM and clears the data RAM
7-1	NU1	R	0h	TI reserved
0	GPADC_TRIGGER		0h	Generates a single cycle pulse to trigger the IFM mode



### 21.5.1.3 REG2 Register (Offset = 8h) [reset = 0h]

REG2 is shown in [Figure 21-4](#) and described in [Table 21-4](#).

Return to the [Table 21-1](#).

gpadc config for IFM

**Figure 21-4. REG2 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONFIG_VALUE_IFM																															
R/W-0h																															

**Table 21-4. REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CONFIG_VALUE_IFM	R/W	0h	Configuration value to be passed to analog in IFM mode

#### 21.5.1.4 REG3 Register (Offset = Ch) [reset = 0h]

REG3 is shown in [Figure 21-5](#) and described in [Table 21-5](#).

Return to the [Table 21-1](#).

gpadc param, skip samples and collect samples for IFM

**Figure 21-5. REG3 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU									SKIP_SAMPLES_IFM						
R-0h									R/W-0h						
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COLLECT_SAMPLES_IFM									PARAM_VAL_IFM						
R/W-0h									R/W-0h						

**Table 21-5. REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-23	NU	R	0h	
22-16	SKIP_SAMPLES_IFM	R/W	0h	number of GPADC clocks to skip after trigger . Number of samples to skip = skip_samples_ifm[3:0]x(2skip_samples_ifm[6:4])
15-8	COLLECT_SAMPLES_IFM	R/W	0h	number of GPADC readings to collect
7-0	PARAM_VAL_IFM	R/W	0h	Param value to be passed to analog in IFM mode(after one hot encoding)

### 21.5.1.5 REG4 Register (Offset = 10h) [reset = 0h]

REG4 is shown in [Figure 21-6](#) and described in [Table 21-6](#).

Return to the [Table 21-1](#).

Base address for Chirp profile 0 in instruction packet RAM

**Figure 21-6. REG4 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PKT_RAM_BASE_ADDR_CP3								PKT_RAM_BASE_ADDR_CP2							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKT_RAM_BASE_ADDR_CP1								PKT_RAM_BASE_ADDR_CP0							
R/W-0h								R/W-0h							

**Table 21-6. REG4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PKT_RAM_BASE_ADDR_CP3	R/W	0h	TI reserved
23-16	PKT_RAM_BASE_ADDR_CP2	R/W	0h	TI reserved
15-8	PKT_RAM_BASE_ADDR_CP1	R/W	0h	(End-Address + 1) of instruction-ram in CTM mode
7-0	PKT_RAM_BASE_ADDR_CP0	R/W	0h	Start Address of instruction-ram in CTM mode

### 21.5.1.6 REG5 Register (Offset = 14h) [reset = 0h]

REG5 is shown in [Figure 21-7](#) and described in [Table 21-7](#).

Return to the [Table 21-1](#).

Base address for Chirp profile 1 in instruction packet RAM

**Figure 21-7. REG5 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PKT_RAM_BASE_ADDR_CP7								PKT_RAM_BASE_ADDR_CP6							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKT_RAM_BASE_ADDR_CP5								PKT_RAM_BASE_ADDR_CP4							
R/W-0h								R/W-0h							

**Table 21-7. REG5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PKT_RAM_BASE_ADDR_CP7	R/W	0h	TI reserved
23-16	PKT_RAM_BASE_ADDR_CP6	R/W	0h	TI reserved
15-8	PKT_RAM_BASE_ADDR_CP5	R/W	0h	TI reserved
7-0	PKT_RAM_BASE_ADDR_CP4	R/W	0h	TI reserved

### 21.5.1.7 REG6 Register (Offset = 18h) [reset = 0h]

REG6 is shown in [Figure 21-8](#) and described in [Table 21-8](#).

Return to the [Table 21-1](#).

Base address for Chirp profile 2 in instruction packet RAM

**Figure 21-8. REG6 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PKT_RAM_BASE_ADDR_CP11								PKT_RAM_BASE_ADDR_CP10							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKT_RAM_BASE_ADDR_CP9								PKT_RAM_BASE_ADDR_CP8							
R/W-0h								R/W-0h							

**Table 21-8. REG6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PKT_RAM_BASE_ADDR_CP11	R/W	0h	TI reserved
23-16	PKT_RAM_BASE_ADDR_CP10	R/W	0h	TI reserved
15-8	PKT_RAM_BASE_ADDR_CP9	R/W	0h	TI reserved
7-0	PKT_RAM_BASE_ADDR_CP8	R/W	0h	TI reserved

### 21.5.1.8 REG7 Register (Offset = 1Ch) [reset = 0h]

REG7 is shown in [Figure 21-9](#) and described in [Table 21-9](#).

Return to the [Table 21-1](#).

Base address for Chirp profile 3 in instruction packet RAM

**Figure 21-9. REG7 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
PKT_RAM_BASE_ADDR_CP15								PKT_RAM_BASE_ADDR_CP14							
R/W-0h								R/W-0h							
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PKT_RAM_BASE_ADDR_CP13								PKT_RAM_BASE_ADDR_CP12							
R/W-0h								R/W-0h							

**Table 21-9. REG7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	PKT_RAM_BASE_ADDR_CP15	R/W	0h	TI reserved
23-16	PKT_RAM_BASE_ADDR_CP14	R/W	0h	TI reserved
15-8	PKT_RAM_BASE_ADDR_CP13	R/W	0h	TI reserved
7-0	PKT_RAM_BASE_ADDR_CP12	R/W	0h	TI reserved

### 21.5.1.9 REG8 Register (Offset = 20h) [reset = 0h]

REG8 is shown in [Figure 21-10](#) and described in [Table 21-10](#).

Return to the [Table 21-1](#).

**Figure 21-10. REG8 Register**

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							GPADC_CLK_ENABLE
R-0h							R/W-0h
7	6	5	4	3	2	1	0
GPADC_CLK_DIV							
R/W-0h							

**Table 21-10. REG8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	NU	R	0h	
8	GPADC_CLK_ENABLE	R/W	0h	TI reserved
7-0	GPADC_CLK_DIV	R/W	0h	TI reserved

### 21.5.1.10 REG9 Register (Offset = 24h) [reset = 0h]

REG9 is shown in [Figure 21-11](#) and described in [Table 21-11](#).

Return to the [Table 21-1](#).

**Figure 21-11. REG9 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARAM_NOT_USED_TX_ENA1_OFF																															
R/W-0h																															

**Table 21-11. REG9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PARAM_NOT_USED_TX_ENA1_OFF	R/W	0h	TI reserved



### 21.5.1.11 REG10 Register (Offset = 28h) [reset = 0h]

REG10 is shown in [Figure 21-12](#) and described in [Table 21-12](#).

Return to the [Table 21-1](#).

**Figure 21-12. REG10 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARAM_NOT_USED_TX_ENA2_OFF																															
R/W-0h																															

**Table 21-12. REG10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PARAM_NOT_USED_TX_ENA2_OFF	R/W	0h	TI reserved

### 21.5.1.12 REG11 Register (Offset = 2Ch) [reset = 0h]

REG11 is shown in [Figure 21-13](#) and described in [Table 21-13](#).

Return to the [Table 21-1](#).

**Figure 21-13. REG11 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARAM_NOT_USED_TX_ENA3_OFF																															
R/W-0h																															

**Table 21-13. REG11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	PARAM_NOT_USED_TX_ENA3_OFF	R/W	0h	TI reserved

### 21.5.1.13 REG12 Register (Offset = 30h) [reset = 0h]

REG12 is shown in [Figure 21-14](#) and described in [Table 21-14](#).

Return to the [Table 21-1](#).

**Figure 21-14. REG12 Register**

31	30	29	28	27	26	25	24
DRAM_REPAIRED_BIT							
R-0h							
23	22	21	20	19	18	17	16
DRAM_ECC_ERR_ADDR							
R-0h							
15	14	13	12	11	10	9	8
NU2						DRAM_ECC_ERR_CLR	
R-0h						0h	
7	6	5	4	3	2	1	0
NU1						DRAM_ECC_ENABLE	
R-0h						R/W-0h	

**Table 21-14. REG12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	DRAM_REPAIRED_BIT	R	0h	TI reserved
23-16	DRAM_ECC_ERR_ADDR	R	0h	TI reserved
15-9	NU2	R	0h	TI reserved
8	DRAM_ECC_ERR_CLR		0h	TI reserved
7-1	NU1	R	0h	TI reserved
0	DRAM_ECC_ENABLE	R/W	0h	

#### 21.5.1.14 REG13 Register (Offset = 34h) [reset = 0h]

REG13 is shown in [Figure 21-15](#) and described in [Table 21-15](#).

Return to the [Table 21-1](#).

**Figure 21-15. REG13 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE_WR2																															
R/W-0h																															

**Table 21-15. REG13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SPARE_WR2	R/W	0h	TI reserved

### 21.5.1.15 REG14 Register (Offset = 38h) [reset = 0h]

REG14 is shown in [Figure 21-16](#) and described in [Table 21-16](#).

Return to the [Table 21-1](#).

Sum of GP ADC readings

**Figure 21-16. REG14 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU												SUM_IFM																			
R-0h												R-0h																			

**Table 21-16. REG14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-20	NU	R	0h	TI reserved
19-0	SUM_IFM	R	0h	Sum of GP ADC readings

### 21.5.1.16 REG15 Register (Offset = 3Ch) [reset = 0h]

REG15 is shown in [Figure 21-17](#) and described in [Table 21-17](#).

Return to the [Table 21-1](#).

Min and Max of GP ADC readings

**Figure 21-17. REG15 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NU2						MAX_GPADC										NU1						MIN_GPADC									
R-0h						R-0h										R-0h						R-0h									

**Table 21-17. REG15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-26	NU2	R	0h	TI reserved
25-16	MAX_GPADC	R	0h	Max of GPADC readings
15-10	NU1	R	0h	TI reserved
9-0	MIN_GPADC	R	0h	Min of GPADC readings

### 21.5.1.17 REG16 Register (Offset = 40h) [reset = 0h]

REG16 is shown in [Figure 21-18](#) and described in [Table 21-18](#).

Return to the [Table 21-1](#).

**Figure 21-18. REG16 Register**

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							GPADC_MEM_I NIT_DONE_ST AT
R-0h							R-0h

**Table 21-18. REG16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	TI reserved
0	GPADC_MEM_INIT_DONE_STAT	R	0h	Status for Data Mem init done.Used for FW polling .Will read '0' when init process is under progress

### 21.5.1.18 REG17 Register (Offset = 44h) [reset = 0h]

REG17 is shown in [Figure 21-19](#) and described in [Table 21-19](#).

Return to the [Table 21-1](#).

**Figure 21-19. REG17 Register**

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							GPADC_IFM_D ONE_STATUS
R-0h							R-0h

**Table 21-19. REG17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	TI reserved
0	GPADC_IFM_DONE_STA TUS	R	0h	Test completion status in IFM mode.Used for FW polling



### 21.5.1.19 REG18 Register (Offset = 48h) [reset = 0h]

REG18 is shown in [Figure 21-20](#) and described in [Table 21-20](#).

Return to the [Table 21-1](#).

**Figure 21-20. REG18 Register**

31	30	29	28	27	26	25	24
NU							
R-0h							
23	22	21	20	19	18	17	16
NU							
R-0h							
15	14	13	12	11	10	9	8
NU							
R-0h							
7	6	5	4	3	2	1	0
NU							GPADC_IFM_D ONE_CLR
R-0h							0h

**Table 21-20. REG18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-1	NU	R	0h	TI reserved
0	GPADC_IFM_DONE_CLR		0h	Clear "ifm_done_status"

### 21.5.1.20 REG19 Register (Offset = 4Ch) [reset = 0h]

REG19 is shown in [Figure 21-21](#) and described in [Table 21-21](#).

Return to the [Table 21-1](#).

**Figure 21-21. REG19 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
NU															
R-0h															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPADC_SAMPLES_FRAME															
R-0h															

**Table 21-21. REG19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-16	NU	R	0h	TI reserved
15-0	GPADC_SAMPLES_FRAME	R	0h	Total number of GPADC samples collected in a frame

### 21.5.1.21 REG20 Register (Offset = 50h) [reset = 0h]

REG20 is shown in [Figure 21-22](#) and described in [Table 21-22](#).

Return to the [Table 21-1](#).

**Figure 21-22. REG20 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE_RD1																															
R-0h																															

**Table 21-22. REG20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SPARE_RD1	R	0h	TI reserved

### 21.5.1.22 REG21 Register (Offset = 54h) [reset = 0h]

REG21 is shown in [Figure 21-23](#) and described in [Table 21-23](#).

Return to the [Table 21-1](#).

**Figure 21-23. REG21 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE_RD2																															
R-0h																															

**Table 21-23. REG21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SPARE_RD2	R	0h	TI reserved

### 21.5.1.23 REG22 Register (Offset = 58h) [reset = 0h]

REG22 is shown in [Figure 21-24](#) and described in [Table 21-24](#).

Return to the [Table 21-1](#).

**Figure 21-24. REG22 Register**

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPARE_WR1																															
R/W-0h																															

**Table 21-24. REG22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	SPARE_WR1	R/W	0h	TI reserved

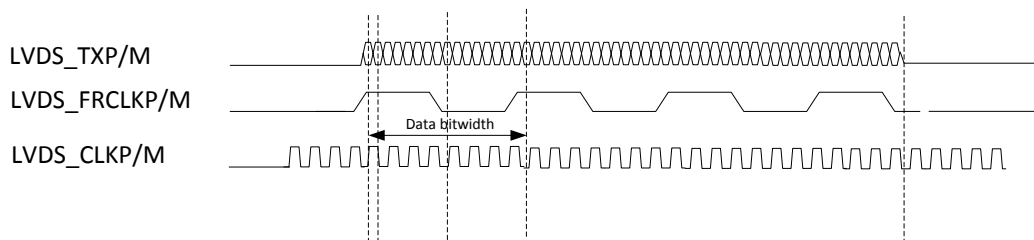


22.1 LVDS Overview.....	5209
22.2 LVDS Programming Sequence.....	5209
22.3 CBUFF and LVDS Registers.....	5211

## 22.1 LVDS Overview

The LVDS interface includes the following signals:

- LVDS bit clock
- LVDS data lanes (the HSI integration chapter specific to the device lists the number of available lanes specific to the device)
- LVDS frame clock
- LVDS data\_valid signal



**Figure 22-1. LVDS Interface Timings**

The LVDS interface supports the following data rates(only DDR mode is supported):

- 900 Mbps (450-MHz DDR Clock)
- 600 Mbps (300-MHz DDR Clock)
- 450 Mbps (225-MHz DDR Clock)
- 400 Mbps (200-MHz DDR Clock)
- 300 Mbps (150-MHz DDR Clock)
- 225 Mbps (112.5-MHz DDR Clock)
- 150 Mbps (75-MHz DDR Clock)

Refer to the device data sheet for more details.

## 22.2 LVDS Programming Sequence

The following sections show the programming sequence needed before the hardware triggers are generated to initiate the high-speed LVDS data transmission.

### 22.2.1 LVDS Global Initialization

**Table 22-1. Main Sequence – TOP\_CTRL and Global Configuration**

Steps	Register/Bit Field/Programming	Value
Power on the LVDS I/Os	MSS_TOP_CTRL.LVDS_PAD_CTRL0 MSS_TOP_CTRL.LVDS_PAD_CTRL1	0x0 0x0
Power off the LVDS I/Os	MSS_TOP_CTRL.LVDS_PAD_CTRL0 MSS_TOP_CTRL.LVDS_PAD_CTRL1	0x39393939 0x01003939
Power on the LVDS I/Os	MSS_TOP_CTRL.LVDS_PAD_CTRL0 MSS_TOP_CTRL.LVDS_PAD_CTRL1	0x0 0x0

### 22.2.2 CBUFF Configuration

**Table 22-2. Main Sequence – CBUFF LVDS Static Configuration**

Steps	Register/Bit Field/Programming	Value
Assert the CBUFF soft reset	CONFIG_REG_0.CSWCRST	0x1
Configure CBUFF for LVDS data transfer	CONFIG_REG_0.CFG_1LVDS_0CSI	0x1

**Table 22-2. Main Sequence – CBUFF LVDS Static Configuration (continued)**

Steps	Register/Bit Field/Programming	Value
Configure static values for LVDS	CONFIG_REG_0.CVC0EN	
	CFG_SPHDR_ADDRESS	0x55555555
	CFG_CMD_VSVAL	0x55555555
	CFG_CMD_VEVAL	0xAAAAAAAA
	CFG_LPHDR_ADDRESS	0xAAAAAAAA
	CFG_LVDS_GEN_0.CCSMEN	0x1
Configure the number of chirps in a frame	CFG_CHIRPS_PER_FRAME	X
Configure static values for LVDS based on LVDS CRC enabled or disabled	CFG_LVDS_GEN_0.CBCRCEN CFG_CMD_HEVAL CFG_CMD_HSVAL	X
Enable the LVDS lanes	CFG_LVDS_GEN_0.CFG_LVDS_LANE[X]_EN	0x1
Configure the alignment for start of samples	CFG_LVDS_GEN_0.CPOSSEL	0x-
Configure the LVDS FIFO initial threshold	CFG_LVDS_GEN_0.CFDLY	0x8
Set the 3C3L mode if the system configuration is interleaved 3 channel – 3 lane	CFG_LVDS_GEN_1.C3C3L	0x-
Configure the lane-mapping format registers	CFG_LVDS_MAPPING_LANE[X]_FMT_0 CFG_LVDS_MAPPING_LANE[X]_FMT_1	X
Release the CBUFF from soft reset	CONFIG_REG_0.CSWCRST	0x0

The configuration in [Table 22-3](#) should be performed for each linklist entry required to transmit the LVDS packet.

**Table 22-3. Main Sequence – CBUFF Linklist**

Steps	Register/Bit Field/Programming	Value
Set the valid for the linklist	CFG_DATA_LL[X].LL[X]_VALID	0x1
If the linklist is the start of a new LVDS packet	CFG_DATA_LL[X].LL[X]_LPHDR_EN CFG_DATA_LL[X].LL[X]_HS	0x-
Configure the long packet header to static value for LVDS	CFG_DATA_LL[X]_LPHDR_VAL	0xBBBBBBBB
If the linklist is the end of a LVDS packet	CFG_DATA_LL[X].LL[X]_HE	0x-
Configure the size in CBUFF units	CFG_DATA_LL[X].LL[X]_SIZE	X
Configure the format of the CSI2 packet to which the linklist belongs	CFG_DATA_LL[X].LL[X]_FMT	X
Select the LVDS format-mapping register for the LVDS packet	CFG_DATA_LL[X].LL[X]_FMT_MAP	0x-
Set the input format	CFG_DATA_LL[X].LL[X]_FMT_IN	X
Set the Linklist write threshold	CFG_DATA_LL[X]_THRESHOLD.LL[X]_WR_THRESHOLD	X
Set the Linklist read threshold	CFG_DATA_LL[X]_THRESHOLD.LL[X]_WR_THRESHOLD	X



## 22.3 CBUFF and LVDS Registers

### 22.3.1 DSS\_CBUFF Registers

Table 22-4 lists the memory-mapped registers for the DSS\_CBUFF registers. All register offset addresses not listed in Table 22-4 should be considered as reserved locations and the register contents should not be modified.

**Table 22-4. DSS\_CBUFF Registers**

Offset	Acronym	Register Name	Section
0h	CONFIG_REG_0	CONFIG_REG_0	<a href="#">Go</a>
4h	CFG_SPHDR_ADDRESS	CFG_SPHDR_ADDRESS	<a href="#">Go</a>
8h	CFG_CMD_HSVAL	CFG_CMD_HSVAL	<a href="#">Go</a>
Ch	CFG_CMD_HEVAL	CFG_CMD_HEVAL	<a href="#">Go</a>
10h	CFG_CMD_VSVAL	CFG_CMD_VSVAL	<a href="#">Go</a>
14h	CFG_CMD_VEVAL	CFG_CMD_VEVAL	<a href="#">Go</a>
18h	CFG_LPHDR_ADDRESS	CFG_LPHDR_ADDRESS	<a href="#">Go</a>
20h	CFG_CHIRPS_PER_FRAME	CFG_CHIRPS_PER_FRAME	<a href="#">Go</a>
24h	CFG_FIFO_FREE_THRESHOLD	CFG_FIFO_FREE_THRESHOLD	<a href="#">Go</a>
28h	CFG_LPPYLD_ADDRESS	CFG_LPPYLD_ADDRESS	<a href="#">Go</a>
2Ch	CFG_DELAY_CONFIG	CFG_DELAY_CONFIG	<a href="#">Go</a>
30h	CFG_DATA_LL0	CFG_DATA_LL0	<a href="#">Go</a>
34h	CFG_DATA_LL0_LPHDR_VAL	CFG_DATA_LL0_LPHDR_VAL	<a href="#">Go</a>
38h	CFG_DATA_LL0_THRESHOLD	CFG_DATA_LL0_THRESHOLD	<a href="#">Go</a>
3Ch	CFG_DATA_LL1	CFG_DATA_LL1	<a href="#">Go</a>
40h	CFG_DATA_LL1_LPHDR_VAL	CFG_DATA_LL1_LPHDR_VAL	<a href="#">Go</a>
44h	CFG_DATA_LL1_THRESHOLD	CFG_DATA_LL1_THRESHOLD	<a href="#">Go</a>
48h	CFG_DATA_LL2	CFG_DATA_LL2	<a href="#">Go</a>
4Ch	CFG_DATA_LL2_LPHDR_VAL	CFG_DATA_LL2_LPHDR_VAL	<a href="#">Go</a>
50h	CFG_DATA_LL2_THRESHOLD	CFG_DATA_LL2_THRESHOLD	<a href="#">Go</a>
54h	CFG_DATA_LL3	CFG_DATA_LL3	<a href="#">Go</a>
58h	CFG_DATA_LL3_LPHDR_VAL	CFG_DATA_LL3_LPHDR_VAL	<a href="#">Go</a>
5Ch	CFG_DATA_LL3_THRESHOLD	CFG_DATA_LL3_THRESHOLD	<a href="#">Go</a>
60h	CFG_DATA_LL4	CFG_DATA_LL4	<a href="#">Go</a>
64h	CFG_DATA_LL4_LPHDR_VAL	CFG_DATA_LL4_LPHDR_VAL	<a href="#">Go</a>
68h	CFG_DATA_LL4_THRESHOLD	CFG_DATA_LL4_THRESHOLD	<a href="#">Go</a>
6Ch	CFG_DATA_LL5	CFG_DATA_LL5	<a href="#">Go</a>
70h	CFG_DATA_LL5_LPHDR_VAL	CFG_DATA_LL5_LPHDR_VAL	<a href="#">Go</a>
74h	CFG_DATA_LL5_THRESHOLD	CFG_DATA_LL5_THRESHOLD	<a href="#">Go</a>
78h	CFG_DATA_LL6	CFG_DATA_LL6	<a href="#">Go</a>
7Ch	CFG_DATA_LL6_LPHDR_VAL	CFG_DATA_LL6_LPHDR_VAL	<a href="#">Go</a>
80h	CFG_DATA_LL6_THRESHOLD	CFG_DATA_LL6_THRESHOLD	<a href="#">Go</a>
84h	CFG_DATA_LL7	CFG_DATA_LL7	<a href="#">Go</a>
88h	CFG_DATA_LL7_LPHDR_VAL	CFG_DATA_LL7_LPHDR_VAL	<a href="#">Go</a>
8Ch	CFG_DATA_LL7_THRESHOLD	CFG_DATA_LL7_THRESHOLD	<a href="#">Go</a>
90h	CFG_DATA_LL8	CFG_DATA_LL8	<a href="#">Go</a>
94h	CFG_DATA_LL8_LPHDR_VAL	CFG_DATA_LL8_LPHDR_VAL	<a href="#">Go</a>
98h	CFG_DATA_LL8_THRESHOLD	CFG_DATA_LL8_THRESHOLD	<a href="#">Go</a>
9Ch	CFG_DATA_LL9	CFG_DATA_LL9	<a href="#">Go</a>
A0h	CFG_DATA_LL9_LPHDR_VAL	CFG_DATA_LL9_LPHDR_VAL	<a href="#">Go</a>
A4h	CFG_DATA_LL9_THRESHOLD	CFG_DATA_LL9_THRESHOLD	<a href="#">Go</a>

**Table 22-4. DSS\_CBUFF Registers (continued)**

Offset	Acronym	Register Name	Section
A8h	CFG_DATA_LL10	CFG_DATA_LL10	<a href="#">Go</a>
ACh	CFG_DATA_LL10_LPHDR_VAL	CFG_DATA_LL10_LPHDR_VAL	<a href="#">Go</a>
B0h	CFG_DATA_LL10_THRESHOLD	CFG_DATA_LL10_THRESHOLD	<a href="#">Go</a>
B4h	CFG_DATA_LL11	CFG_DATA_LL11	<a href="#">Go</a>
B8h	CFG_DATA_LL11_LPHDR_VAL	CFG_DATA_LL11_LPHDR_VAL	<a href="#">Go</a>
BCh	CFG_DATA_LL11_THRESHOLD	CFG_DATA_LL11_THRESHOLD	<a href="#">Go</a>
C0h	CFG_DATA_LL12	CFG_DATA_LL12	<a href="#">Go</a>
C4h	CFG_DATA_LL12_LPHDR_VAL	CFG_DATA_LL12_LPHDR_VAL	<a href="#">Go</a>
C8h	CFG_DATA_LL12_THRESHOLD	CFG_DATA_LL12_THRESHOLD	<a href="#">Go</a>
CCh	CFG_DATA_LL13	CFG_DATA_LL13	<a href="#">Go</a>
D0h	CFG_DATA_LL13_LPHDR_VAL	CFG_DATA_LL13_LPHDR_VAL	<a href="#">Go</a>
D4h	CFG_DATA_LL13_THRESHOLD	CFG_DATA_LL13_THRESHOLD	<a href="#">Go</a>
D8h	CFG_DATA_LL14	CFG_DATA_LL14	<a href="#">Go</a>
DCh	CFG_DATA_LL14_LPHDR_VAL	CFG_DATA_LL14_LPHDR_VAL	<a href="#">Go</a>
E0h	CFG_DATA_LL14_THRESHOLD	CFG_DATA_LL14_THRESHOLD	<a href="#">Go</a>
E4h	CFG_DATA_LL15	CFG_DATA_LL15	<a href="#">Go</a>
E8h	CFG_DATA_LL15_LPHDR_VAL	CFG_DATA_LL15_LPHDR_VAL	<a href="#">Go</a>
ECh	CFG_DATA_LL15_THRESHOLD	CFG_DATA_LL15_THRESHOLD	<a href="#">Go</a>
F0h	CFG_DATA_LL16	CFG_DATA_LL16	<a href="#">Go</a>
F4h	CFG_DATA_LL16_LPHDR_VAL	CFG_DATA_LL16_LPHDR_VAL	<a href="#">Go</a>
F8h	CFG_DATA_LL16_THRESHOLD	CFG_DATA_LL16_THRESHOLD	<a href="#">Go</a>
FCh	CFG_DATA_LL17	CFG_DATA_LL17	<a href="#">Go</a>
100h	CFG_DATA_LL17_LPHDR_VAL	CFG_DATA_LL17_LPHDR_VAL	<a href="#">Go</a>
104h	CFG_DATA_LL17_THRESHOLD	CFG_DATA_LL17_THRESHOLD	<a href="#">Go</a>
108h	CFG_DATA_LL18	CFG_DATA_LL18	<a href="#">Go</a>
10Ch	CFG_DATA_LL18_LPHDR_VAL	CFG_DATA_LL18_LPHDR_VAL	<a href="#">Go</a>
110h	CFG_DATA_LL18_THRESHOLD	CFG_DATA_LL18_THRESHOLD	<a href="#">Go</a>
114h	CFG_DATA_LL19	CFG_DATA_LL19	<a href="#">Go</a>
118h	CFG_DATA_LL19_LPHDR_VAL	CFG_DATA_LL19_LPHDR_VAL	<a href="#">Go</a>
11Ch	CFG_DATA_LL19_THRESHOLD	CFG_DATA_LL19_THRESHOLD	<a href="#">Go</a>
120h	CFG_DATA_LL20	CFG_DATA_LL20	<a href="#">Go</a>
124h	CFG_DATA_LL20_LPHDR_VAL	CFG_DATA_LL20_LPHDR_VAL	<a href="#">Go</a>
128h	CFG_DATA_LL20_THRESHOLD	CFG_DATA_LL20_THRESHOLD	<a href="#">Go</a>
12Ch	CFG_DATA_LL21	CFG_DATA_LL21	<a href="#">Go</a>
130h	CFG_DATA_LL21_LPHDR_VAL	CFG_DATA_LL21_LPHDR_VAL	<a href="#">Go</a>
134h	CFG_DATA_LL21_THRESHOLD	CFG_DATA_LL21_THRESHOLD	<a href="#">Go</a>
138h	CFG_DATA_LL22	CFG_DATA_LL22	<a href="#">Go</a>
13Ch	CFG_DATA_LL22_LPHDR_VAL	CFG_DATA_LL22_LPHDR_VAL	<a href="#">Go</a>
140h	CFG_DATA_LL22_THRESHOLD	CFG_DATA_LL22_THRESHOLD	<a href="#">Go</a>
144h	CFG_DATA_LL23	CFG_DATA_LL23	<a href="#">Go</a>
148h	CFG_DATA_LL23_LPHDR_VAL	CFG_DATA_LL23_LPHDR_VAL	<a href="#">Go</a>
14Ch	CFG_DATA_LL23_THRESHOLD	CFG_DATA_LL23_THRESHOLD	<a href="#">Go</a>
150h	CFG_DATA_LL24	CFG_DATA_LL24	<a href="#">Go</a>
154h	CFG_DATA_LL24_LPHDR_VAL	CFG_DATA_LL24_LPHDR_VAL	<a href="#">Go</a>
158h	CFG_DATA_LL24_THRESHOLD	CFG_DATA_LL24_THRESHOLD	<a href="#">Go</a>

**Table 22-4. DSS\_CBUFF Registers (continued)**

Offset	Acronym	Register Name	Section
15Ch	CFG_DATA_LL25	CFG_DATA_LL25	<a href="#">Go</a>
160h	CFG_DATA_LL25_LPHDR_VAL	CFG_DATA_LL25_LPHDR_VAL	<a href="#">Go</a>
164h	CFG_DATA_LL25_THRESHOLD	CFG_DATA_LL25_THRESHOLD	<a href="#">Go</a>
168h	CFG_DATA_LL26	CFG_DATA_LL26	<a href="#">Go</a>
16Ch	CFG_DATA_LL26_LPHDR_VAL	CFG_DATA_LL26_LPHDR_VAL	<a href="#">Go</a>
170h	CFG_DATA_LL26_THRESHOLD	CFG_DATA_LL26_THRESHOLD	<a href="#">Go</a>
174h	CFG_DATA_LL27	CFG_DATA_LL27	<a href="#">Go</a>
178h	CFG_DATA_LL27_LPHDR_VAL	CFG_DATA_LL27_LPHDR_VAL	<a href="#">Go</a>
17Ch	CFG_DATA_LL27_THRESHOLD	CFG_DATA_LL27_THRESHOLD	<a href="#">Go</a>
180h	CFG_DATA_LL28	CFG_DATA_LL28	<a href="#">Go</a>
184h	CFG_DATA_LL28_LPHDR_VAL	CFG_DATA_LL28_LPHDR_VAL	<a href="#">Go</a>
188h	CFG_DATA_LL28_THRESHOLD	CFG_DATA_LL28_THRESHOLD	<a href="#">Go</a>
18Ch	CFG_DATA_LL29	CFG_DATA_LL29	<a href="#">Go</a>
190h	CFG_DATA_LL29_LPHDR_VAL	CFG_DATA_LL29_LPHDR_VAL	<a href="#">Go</a>
194h	CFG_DATA_LL29_THRESHOLD	CFG_DATA_LL29_THRESHOLD	<a href="#">Go</a>
198h	CFG_DATA_LL30	CFG_DATA_LL30	<a href="#">Go</a>
19Ch	CFG_DATA_LL30_LPHDR_VAL	CFG_DATA_LL30_LPHDR_VAL	<a href="#">Go</a>
1A0h	CFG_DATA_LL30_THRESHOLD	CFG_DATA_LL30_THRESHOLD	<a href="#">Go</a>
1A4h	CFG_DATA_LL31	CFG_DATA_LL31	<a href="#">Go</a>
1A8h	CFG_DATA_LL31_LPHDR_VAL	CFG_DATA_LL31_LPHDR_VAL	<a href="#">Go</a>
1ACh	CFG_DATA_LL31_THRESHOLD	CFG_DATA_LL31_THRESHOLD	<a href="#">Go</a>
1B0h	CFG_LVDS_MAPPING_LANE0_FMT_0	CFG_LVDS_MAPPING_LANE0_FMT_0	<a href="#">Go</a>
1B4h	CFG_LVDS_MAPPING_LANE1_FMT_0	CFG_LVDS_MAPPING_LANE1_FMT_0	<a href="#">Go</a>
1B8h	CFG_LVDS_MAPPING_LANE2_FMT_0	CFG_LVDS_MAPPING_LANE2_FMT_0	<a href="#">Go</a>
1BCh	CFG_LVDS_MAPPING_LANE3_FMT_0	CFG_LVDS_MAPPING_LANE3_FMT_0	<a href="#">Go</a>
1C0h	CFG_LVDS_MAPPING_LANE0_FMT_1	CFG_LVDS_MAPPING_LANE0_FMT_1	<a href="#">Go</a>
1C4h	CFG_LVDS_MAPPING_LANE1_FMT_1	CFG_LVDS_MAPPING_LANE1_FMT_1	<a href="#">Go</a>
1C8h	CFG_LVDS_MAPPING_LANE2_FMT_1	CFG_LVDS_MAPPING_LANE2_FMT_1	<a href="#">Go</a>
1CCh	CFG_LVDS_MAPPING_LANE3_FMT_1	CFG_LVDS_MAPPING_LANE3_FMT_1	<a href="#">Go</a>
1D0h	CFG_LVDS_GEN_0	CFG_LVDS_GEN_0	<a href="#">Go</a>
1D4h	CFG_LVDS_GEN_1	CFG_LVDS_GEN_1	<a href="#">Go</a>
1D8h	CFG_LVDS_GEN_2	CFG_LVDS_GEN_2	<a href="#">Go</a>
1DCh	CFG_MASK_REG0	CFG_MASK_REG0	<a href="#">Go</a>
1E0h	CFG_MASK_REG1	CFG_MASK_REG1	<a href="#">Go</a>
1E4h	CFG_MASK_REG2	CFG_MASK_REG2	<a href="#">Go</a>
1E8h	CFG_MASK_REG3	CFG_MASK_REG3	<a href="#">Go</a>
1ECh	STAT_CBUFF_REG0	STAT_CBUFF_REG0	<a href="#">Go</a>
1F0h	STAT_CBUFF_REG1	STAT_CBUFF_REG1	<a href="#">Go</a>
1F4h	STAT_CBUFF_REG2	STAT_CBUFF_REG2	<a href="#">Go</a>
1F8h	STAT_CBUFF_REG3	STAT_CBUFF_REG3	<a href="#">Go</a>
1FCh	STAT_LVDS_REG0	STAT_LVDS_REG0	<a href="#">Go</a>
200h	STAT_LVDS_REG1	STAT_LVDS_REG1	<a href="#">Go</a>
204h	STAT_LVDS_REG2	STAT_LVDS_REG2	<a href="#">Go</a>
208h	STAT_LVDS_REG3	STAT_LVDS_REG3	<a href="#">Go</a>
20Ch	CLR_CBUFF_REG0	CLR_CBUFF_REG0	<a href="#">Go</a>

**Table 22-4. DSS\_CBUFF Registers (continued)**

Offset	Acronym	Register Name	Section
210h	CLR_CBUFF_REG1	CLR_CBUFF_REG1	<a href="#">Go</a>
214h	CLR_LVDS_REG0	CLR_LVDS_REG0	<a href="#">Go</a>
218h	CLR_LVDS_REG1	CLR_LVDS_REG1	<a href="#">Go</a>
21Ch	STAT_CBUFF_ECC_REG	STAT_CBUFF_ECC_REG	<a href="#">Go</a>
220h	MASK_CBUFF_ECC_REG	MASK_CBUFF_ECC_REG	<a href="#">Go</a>
224h	CLR_CBUFF_ECC_REG	CLR_CBUFF_ECC_REG	<a href="#">Go</a>
228h	STAT_SAFETY	STAT_SAFETY	<a href="#">Go</a>
22Ch	MASK_SAFETY	MASK_SAFETY	<a href="#">Go</a>
230h	CLR_SAFETY	CLR_SAFETY	<a href="#">Go</a>

Complex bit access types are encoded to fit into small table cells. [Table 22-5](#) shows the codes that are used for access types in this section.

**Table 22-5. DSS\_CBUFF Access Type Codes**

Access Type	Code	Description
Read Type		
R	R	Read
Write Type		
W	W	Write
Reset or Default Value		
-n		Value after reset or the default value

### 22.3.1.1 CONFIG\_REG\_0 Register (Offset = 0h) [Reset = 00000000h]

CONFIG\_REG\_0 is shown in [Table 22-6](#).

Return to the [Summary Table](#).

Basic Config register

**Table 22-6. CONFIG\_REG\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	dbussel	R/W	0h	TI Internal feature. 1 : This selects the debug bus mode transmission on LVDS
27	cswcrst	R/W	0h	CBUFF controller SW Reset 1 => RESET the CBUFF Controller 0 => RELEASE RESET for CBUFF Controller
26	cswlrst	R/W	0h	TI Internal Feature. LVDS logic SW Reset. Debug feature. 1 => RESET the FSM 0 => RELEASE RESET
25	CFG_FRAME_START_TRIG	SW Trigger generation : Write 1 to this bit to generate a Frame Start SW Trigger	0h	TI Internal Feature. Only required for 900 Mbps 4 lane transmission CSI2 only Programming : 0xA : Burst Enable. Set this only for transmission at 900 Mbps Others : Burst disable.
24	CFG_CHIRP_AVAIL_TRIGGER	SW Trigger generation : Write 1 to this bit to generate a Chirp Available SW Trigger		
23-20	CFG_VBUSP_BURST_EN	R/W		
19	dbusen	R/W	0h	TC2 Mode selection. TI Internal feature. 0 : Normal 1 : When in TC2 mode, setting this bit will enable debug bus to sent via LVDS
18	ccfwpen	R/W	0h	TI Internal Feature. Debug only. CSI2 only Programming : CFG_CSI2_FIFO_WORDS_PROCESSING_EN 0 : Use the fifo_free_words directly from CSI2 by vbusp_mstr to decide how many more words to send. 1 : Process the fifo_free_words and use it by vbusp_mstr to decide how many more words to send.
17-16	cvc3en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 3 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 3 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 3 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 3 is generated at end of Frame

**Table 22-6. CONFIG\_REG\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
15-14	cvc2en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 2 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 2 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 2 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 2 is generated at end of Frame
13-12	cvc1en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 1 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 1 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 1 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 1 is generated at end of Frame
11-10	cvc0en	R/W	0h	CSI2 only Programming : 0 : No Vsync packet is sent at Frame boundary 1 : A VSYNC Start packet on Virtual Channel 0 is generated at beginning of Frame 2 : A VSYNC End packet on Virtual Channel 0 is generated at end of Frame 3 : A VSYNC Start packet on Virtual Channel 0 is generated at beginning of Frame and a VSYNC End packet on Virtual Channel 0 is generated at end of Frame
9	crdthsel	R/W	0h	TI Internal Feature. Debug only. CSI2 only Programming : CFG_RDTHRESHOLD_SEL . This is a Debug feature. Not required in Programming model 0 : The read threshold is selected based on the Write Side parsing engine 1 : The read threshold is selected based on the Read Side parsing engine.
8	ccfwlen	R/W	0h	TI Internal Feature. Debug only. CSI2 only Programming : CFG_CSI2_FIFO_WORDS_LOAD_SW_EN. This is a Debug feature. Not required in Programming model When CFG_CSI2_FIFO_WORDS_PROCESSING_EN==1 and CFG_CSI2_FIFO_WORDS_LOAD_SW_EN==1, then a fixed fifo_free_words from CSI2 is not used. Program the CFG_FIFO_FREE_THRESHOLD0 to 0x4
7-4	NU1	R	0h	
3	CFG_SW_TRIG_EN	R/W	0h	Select Chirp Available Trigger Source 0 : Chirp Available trigger will be generated by HW 1 : Chirp Available trigger will be generated by SW
2	cftrigen	R/W	0h	Select Frame Start Trigger Source 0 : Frame trigger will be generated by HW 1 : Frame trigger will be generated by SW
1	CFG_ECC_EN	R/W	0h	0 : Disable ECC on the CBUF FIFO 1 : Enable ECC on the CBUF FIFO
0	CFG_1LVDS_0CSI	R/W	0h	0 : Send data over CSI-2 1 : Send data over LVDS

### 22.3.1.2 CFG\_SPHDR\_ADDRESS Register (Offset = 4h) [Reset = 00000000h]

CFG\_SPHDR\_ADDRESS is shown in [Table 22-7](#).

Return to the [Summary Table](#).

Short Packet Header Address

**Table 22-7. CFG\_SPHDR\_ADDRESS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_SPHDR_ADDRESS	R/W	0h	CSI2 Programming : Configure the CSI_PROTOCOL_ENGINE__CSI_VC_SHORT_PACKET_HEADER Address in the CSI Protocol Engine LVDS Programming : Configure with the static value : 0x55555555



### 22.3.1.3 CFG\_CMD\_HSVAL Register (Offset = 8h) [Reset = 00000000h]

CFG\_CMD\_HSVAL is shown in [Table 22-8](#).

Return to the [Summary Table](#).

HSYNC Value

**Table 22-8. CFG\_CMD\_HSVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_HSVAL	R/W	0h	CSI2 Programming : Configure the HSync Start Short Packet Value LVDS Programming : If LVDS CRC is enabled : Configure with the static value : 0x55555555 If LVDS CRC is disbaled : Configure with the static value : 0xAAAAAAAA

#### 22.3.1.4 CFG\_CMD\_HEVAL Register (Offset = Ch) [Reset = 00000000h]

CFG\_CMD\_HEVAL is shown in [Table 22-9](#).

Return to the [Summary Table](#).

HEND Value

**Table 22-9. CFG\_CMD\_HEVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_HEVAL	R/W	0h	CSI2 Programming : Configure the HSync End Short Packet Value LVDS Programming : If LVDS CRC is enabled : Configure with the static value : 0x33333333 If LVDS CRC is disbaled : Configure with the static value : 0xAAAAAAAA

### 22.3.1.5 CFG\_CMD\_VSVAL Register (Offset = 10h) [Reset = 00000000h]

CFG\_CMD\_VSVAL is shown in [Table 22-10](#).

Return to the [Summary Table](#).

VSYNC Value

**Table 22-10. CFG\_CMD\_VSVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_VSVAL	R/W	0h	CSI2 Programming : Configure the VSync Start Short Packet Value LVDS Programming : Configure with the static value : 0xAAAAAAAA

### 22.3.1.6 CFG\_CMD\_VEVAL Register (Offset = 14h) [Reset = 00000000h]

CFG\_CMD\_VEVAL is shown in [Table 22-11](#).

Return to the [Summary Table](#).

VEND Value

**Table 22-11. CFG\_CMD\_VEVAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_CMD_VEVAL	R/W	0h	CSI2 Programming : Configure the VSync End Short Packet Value LVDS Programming : Configure with the static value : 0xAAAAAAAA

### 22.3.1.7 CFG\_LPHDR\_ADDRESS Register (Offset = 18h) [Reset = 00000000h]

CFG\_LPHDR\_ADDRESS is shown in [Table 22-12](#).

Return to the [Summary Table](#).

Long Packet Address

**Table 22-12. CFG\_LPHDR\_ADDRESS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_LPHDR_ADDRESS	R/W	0h	CSI2 Programming : Configure the CSI_PROTOCOL_ENGINE__CSI_VC_LONG_PACKET_HEADER Address in the CSI Protocol Engine LVDS Programming : Configure with the static value : 0x55555555

### 22.3.1.8 CFG\_CHIRPS\_PER\_FRAME Register (Offset = 20h) [Reset = 00000000h]

CFG\_CHIRPS\_PER\_FRAME is shown in [Table 22-13](#).

Return to the [Summary Table](#).

Number of Chirps per Frame

**Table 22-13. CFG\_CHIRPS\_PER\_FRAME Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_CHIRPS_PER_FRAME	R/W	0h	Configure the number of Chirps in a Frame

### 22.3.1.9 CFG\_FIFO\_FREE\_THRESHOLD Register (Offset = 24h) [Reset = 00000000h]

CFG\_FIFO\_FREE\_THRESHOLD is shown in [Table 22-14](#).

Return to the [Summary Table](#).

CSI2 FIFO threshold for transferring data from CBUFF to CSI2

**Table 22-14. CFG\_FIFO\_FREE\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	CFG_FIFO_FREE_THRESHOLD3	R/W	0h	TI Internal Feature CSI2 only Programming : Configure the threshold used to fill the FIFO3 in the CSI Protocol engine. CBUFF will send data to the Protocol Engine only if there is a larger number of Free slots that that configured in this register. By default, only 1 FIFO will be used so register programming is not Required in Programming model
23-16	CFG_FIFO_FREE_THRESHOLD2	R/W	0h	TI Internal Feature CSI2 only Programming : Configure the threshold used to fill the FIFO2 in the CSI Protocol engine. CBUFF will send data to the Protocol Engine only if there is a larger number of Free slots that that configured in this register. By default, only 1 FIFO will be used so register programming is not Required in Programming model
15-8	CFG_FIFO_FREE_THRESHOLD1	R/W	0h	TI Internal Feature CSI2 only Programming : Configure the threshold used to fill the FIFO1 in the CSI Protocol engine. CBUFF will send data to the Protocol Engine only if there is a larger number of Free slots that that configured in this register. By default, only 1 FIFO will be used so register programming is not Required in Programming model
7-0	CFG_FIFO_FREE_THRESHOLD0	R/W	0h	CSI2 only Programming : Configure the threshold used to fill the FIFO0 in the CSI Protocol engine. CBUFF will send data to the Protocol Engine only if there is a larger number of Free slots that that configured in this register

### 22.3.1.10 CFG\_LPPYLD\_ADDRESS Register (Offset = 28h) [Reset = 00000000h]

CFG\_LPPYLD\_ADDRESS is shown in [Table 22-15](#).

Return to the [Summary Table](#).

Long payload Address

**Table 22-15. CFG\_LPPYLD\_ADDRESS Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_LPPYLD_ADDRESS	R/W	0h	CSI2 only Programming : Configure the CSI_PROTOCOL_ENGINE__CSI_VC_LONG_PACKET_PAYLOAD Address in the CSI Protocol Engine



### 22.3.1.11 CFG\_DELAY\_CONFIG Register (Offset = 2Ch) [Reset = 00000000h]

CFG\_DELAY\_CONFIG is shown in [Table 22-16](#).

Return to the [Summary Table](#).

Delay Config Registers

**Table 22-16. CFG\_DELAY\_CONFIG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-24	NU	R	0h	
23-16	CFG_DATA_WR_DELAY	R/W	0h	TI Internal Feature CSI2 only Programming : Configure an additional delay after sending a Long packet Payload. This is a Debug feature. Not required in Programming model
15-8	CFG_LPHDR_DELAY	R/W	0h	TI Internal Feature CSI2 only Programming : Configure an additional delay after sending a Long packet Header. This is a Debug feature. Not required in Programming model
7-0	CFG_SPHDR_DELAY	R/W	0h	TI Internal Feature CSI2 only Programming : Configure an additional delay after sending a Short packet. This is a Debug feature. Not required in Programming model

### 22.3.1.12 CFG\_DATA\_LL0 Register (Offset = 30h) [Reset = 00000000h]

CFG\_DATA\_LL0 is shown in [Table 22-17](#).

Return to the [Summary Table](#).

Payload Description : Linked list entry 0

**Table 22-17. CFG\_DATA\_LL0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL0_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL0_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL0_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL0_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL0_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL0_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL0_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL0_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL0_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL0_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL0_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL0_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL0_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL0_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL0_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.13 CFG\_DATA\_LL0\_LPHDR\_VAL Register (Offset = 34h) [Reset = 00000000h]

CFG\_DATA\_LL0\_LPHDR\_VAL is shown in [Table 22-18](#).

Return to the [Summary Table](#).

Payload Description : Linked list entry 0

**Table 22-18. CFG\_DATA\_LL0\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL0_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.14 CFG\_DATA\_LL0\_THRESHOLD Register (Offset = 38h) [Reset = 0000X0X0h]

CFG\_DATA\_LL0\_THRESHOLD is shown in [Table 22-19](#).

Return to the [Summary Table](#).

**Table 22-19. CFG\_DATA\_LL0\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll0dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL0_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL0_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.15 CFG\_DATA\_LL1 Register (Offset = 3Ch) [Reset = 00000000h]

CFG\_DATA\_LL1 is shown in [Table 22-20](#).

Return to the [Summary Table](#).

**Table 22-20. CFG\_DATA\_LL1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL1_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL1_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL1_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL1_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL1_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL1_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL1_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL1_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL1_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL1_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL1_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL1_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL1_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL1_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL1_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.16 CFG\_DATA\_LL1\_LPHDR\_VAL Register (Offset = 40h) [Reset = 00000000h]

CFG\_DATA\_LL1\_LPHDR\_VAL is shown in [Table 22-21](#).

Return to the [Summary Table](#).

**Table 22-21. CFG\_DATA\_LL1\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL1_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.17 CFG\_DATA\_LL1\_THRESHOLD Register (Offset = 44h) [Reset = 0000X0X0h]

CFG\_DATA\_LL1\_THRESHOLD is shown in [Table 22-22](#).

Return to the [Summary Table](#).

**Table 22-22. CFG\_DATA\_LL1\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll1dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL1_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL1_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.18 CFG\_DATA\_LL2 Register (Offset = 48h) [Reset = 00000000h]

CFG\_DATA\_LL2 is shown in [Table 22-23](#).

Return to the [Summary Table](#).

**Table 22-23. CFG\_DATA\_LL2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL2_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL2_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL2_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL2_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL2_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL2_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL2_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL2_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL2_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL2_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL2_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL2_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL2_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL2_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL2_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid



### 22.3.1.19 CFG\_DATA\_LL2\_LPHDR\_VAL Register (Offset = 4Ch) [Reset = 00000000h]

CFG\_DATA\_LL2\_LPHDR\_VAL is shown in [Table 22-24](#).

Return to the [Summary Table](#).

**Table 22-24. CFG\_DATA\_LL2\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL2_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.20 CFG\_DATA\_LL2\_THRESHOLD Register (Offset = 50h) [Reset = 0000X0X0h]

CFG\_DATA\_LL2\_THRESHOLD is shown in [Table 22-25](#).

Return to the [Summary Table](#).

**Table 22-25. CFG\_DATA\_LL2\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll2dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL2_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL2_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.21 CFG\_DATA\_LL3 Register (Offset = 54h) [Reset = 00000000h]

CFG\_DATA\_LL3 is shown in [Table 22-26](#).

Return to the [Summary Table](#).

**Table 22-26. CFG\_DATA\_LL3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL3_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL3_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL3_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL3_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL3_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL3_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL3_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL3_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL3_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL3_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL3_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL3_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL3_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL3_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL3_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.22 CFG\_DATA\_LL3\_LPHDR\_VAL Register (Offset = 58h) [Reset = 00000000h]

CFG\_DATA\_LL3\_LPHDR\_VAL is shown in [Table 22-27](#).

Return to the [Summary Table](#).

**Table 22-27. CFG\_DATA\_LL3\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL3_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.23 CFG\_DATA\_LL3\_THRESHOLD Register (Offset = 5Ch) [Reset = 0000X0X0h]

CFG\_DATA\_LL3\_THRESHOLD is shown in [Table 22-28](#).

Return to the [Summary Table](#).

**Table 22-28. CFG\_DATA\_LL3\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll3dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL3_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL3_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.24 CFG\_DATA\_LL4 Register (Offset = 60h) [Reset = 00000000h]

CFG\_DATA\_LL4 is shown in [Table 22-29](#).

Return to the [Summary Table](#).

**Table 22-29. CFG\_DATA\_LL4 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL4_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL4_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL4_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL4_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL4_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL4_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL4_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL4_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL4_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL4_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL4_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL4_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL4_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL4_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL4_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.25 CFG\_DATA\_LL4\_LPHDR\_VAL Register (Offset = 64h) [Reset = 00000000h]

CFG\_DATA\_LL4\_LPHDR\_VAL is shown in [Table 22-30](#).

Return to the [Summary Table](#).

**Table 22-30. CFG\_DATA\_LL4\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL4_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.26 CFG\_DATA\_LL4\_THRESHOLD Register (Offset = 68h) [Reset = 0000X0X0h]

CFG\_DATA\_LL4\_THRESHOLD is shown in [Table 22-31](#).

Return to the [Summary Table](#).

**Table 22-31. CFG\_DATA\_LL4\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll4dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL4_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL4_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model



### 22.3.1.27 CFG\_DATA\_LL5 Register (Offset = 6Ch) [Reset = 00000000h]

CFG\_DATA\_LL5 is shown in [Table 22-32](#).

Return to the [Summary Table](#).

**Table 22-32. CFG\_DATA\_LL5 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL5_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL5_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL5_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL5_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL5_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL5_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL5_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL5_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL5_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL5_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL5_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL5_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL5_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL5_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL5_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.28 CFG\_DATA\_LL5\_LPHDR\_VAL Register (Offset = 70h) [Reset = 00000000h]

CFG\_DATA\_LL5\_LPHDR\_VAL is shown in [Table 22-33](#).

Return to the [Summary Table](#).

**Table 22-33. CFG\_DATA\_LL5\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL5_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.29 CFG\_DATA\_LL5\_THRESHOLD Register (Offset = 74h) [Reset = 0000X0X0h]

CFG\_DATA\_LL5\_THRESHOLD is shown in [Table 22-34](#).

Return to the [Summary Table](#).

**Table 22-34. CFG\_DATA\_LL5\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll5dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL5_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL5_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.30 CFG\_DATA\_LL6 Register (Offset = 78h) [Reset = 00000000h]

CFG\_DATA\_LL6 is shown in [Table 22-35](#).

Return to the [Summary Table](#).

**Table 22-35. CFG\_DATA\_LL6 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL6_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL6_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL6_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL6_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL6_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL6_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL6_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL6_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL6_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL6_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL6_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL6_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL6_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL6_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL6_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.31 CFG\_DATA\_LL6\_LPHDR\_VAL Register (Offset = 7Ch) [Reset = 00000000h]

CFG\_DATA\_LL6\_LPHDR\_VAL is shown in [Table 22-36](#).

Return to the [Summary Table](#).

**Table 22-36. CFG\_DATA\_LL6\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL6_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.32 CFG\_DATA\_LL6\_THRESHOLD Register (Offset = 80h) [Reset = 0000X0X0h]

CFG\_DATA\_LL6\_THRESHOLD is shown in [Table 22-37](#).

Return to the [Summary Table](#).

**Table 22-37. CFG\_DATA\_LL6\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll6dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL6_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL6_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.33 CFG\_DATA\_LL7 Register (Offset = 84h) [Reset = 00000000h]

CFG\_DATA\_LL7 is shown in [Table 22-38](#).

Return to the [Summary Table](#).

**Table 22-38. CFG\_DATA\_LL7 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL7_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL7_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL7_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL7_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL7_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL7_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL7_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL7_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL7_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL7_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL7_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL7_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL7_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL7_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL7_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.34 CFG\_DATA\_LL7\_LPHDR\_VAL Register (Offset = 88h) [Reset = 00000000h]

CFG\_DATA\_LL7\_LPHDR\_VAL is shown in [Table 22-39](#).

Return to the [Summary Table](#).

**Table 22-39. CFG\_DATA\_LL7\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL7_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB



### 22.3.1.35 CFG\_DATA\_LL7\_THRESHOLD Register (Offset = 8Ch) [Reset = 0000X0X0h]

CFG\_DATA\_LL7\_THRESHOLD is shown in [Table 22-40](#).

Return to the [Summary Table](#).

**Table 22-40. CFG\_DATA\_LL7\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll7dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL7_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL7_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.36 CFG\_DATA\_LL8 Register (Offset = 90h) [Reset = 00000000h]

CFG\_DATA\_LL8 is shown in [Table 22-41](#).

Return to the [Summary Table](#).

**Table 22-41. CFG\_DATA\_LL8 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL8_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL8_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL8_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL8_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL8_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL8_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL8_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL8_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL8_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL8_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL8_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL8_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL8_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL8_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL8_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.37 CFG\_DATA\_LL8\_LPHDR\_VAL Register (Offset = 94h) [Reset = 00000000h]

CFG\_DATA\_LL8\_LPHDR\_VAL is shown in [Table 22-42](#).

Return to the [Summary Table](#).

**Table 22-42. CFG\_DATA\_LL8\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL8_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.38 CFG\_DATA\_LL8\_THRESHOLD Register (Offset = 98h) [Reset = 0000X0X0h]

CFG\_DATA\_LL8\_THRESHOLD is shown in [Table 22-43](#).

Return to the [Summary Table](#).

**Table 22-43. CFG\_DATA\_LL8\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll8dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL8_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL8_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.39 CFG\_DATA\_LL9 Register (Offset = 9Ch) [Reset = 00000000h]

CFG\_DATA\_LL9 is shown in [Table 22-44](#).

Return to the [Summary Table](#).

**Table 22-44. CFG\_DATA\_LL9 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL9_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL9_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL9_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL9_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL9_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL9_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL9_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL9_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL9_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL9_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL9_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL9_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL9_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL9_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL9_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.40 CFG\_DATA\_LL9\_LPHDR\_VAL Register (Offset = A0h) [Reset = 00000000h]

CFG\_DATA\_LL9\_LPHDR\_VAL is shown in [Table 22-45](#).

Return to the [Summary Table](#).

**Table 22-45. CFG\_DATA\_LL9\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL9_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.41 CFG\_DATA\_LL9\_THRESHOLD Register (Offset = A4h) [Reset = 0000X0X0h]

CFG\_DATA\_LL9\_THRESHOLD is shown in [Table 22-46](#).

Return to the [Summary Table](#).

**Table 22-46. CFG\_DATA\_LL9\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll9dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL9_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL9_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.42 CFG\_DATA\_LL10 Register (Offset = A8h) [Reset = 00000000h]

CFG\_DATA\_LL10 is shown in [Table 22-47](#).

Return to the [Summary Table](#).

**Table 22-47. CFG\_DATA\_LL10 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL10_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL10_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL10_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL10_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL10_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL10_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL10_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL10_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL10_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL10_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL10_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL10_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL10_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL10_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL10_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid



### 22.3.1.43 CFG\_DATA\_LL10\_LPHDR\_VAL Register (Offset = ACh) [Reset = 00000000h]

CFG\_DATA\_LL10\_LPHDR\_VAL is shown in [Table 22-48](#).

Return to the [Summary Table](#).

**Table 22-48. CFG\_DATA\_LL10\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL10_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.44 CFG\_DATA\_LL10\_THRESHOLD Register (Offset = B0h) [Reset = 0000X0X0h]

CFG\_DATA\_LL10\_THRESHOLD is shown in [Table 22-49](#).

Return to the [Summary Table](#).

**Table 22-49. CFG\_DATA\_LL10\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll10dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL10_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL10_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.45 CFG\_DATA\_LL11 Register (Offset = B4h) [Reset = 00000000h]

CFG\_DATA\_LL11 is shown in [Table 22-50](#).

Return to the [Summary Table](#).

**Table 22-50. CFG\_DATA\_LL11 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL11_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL11_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL11_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL11_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL11_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL11_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL11_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL11_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL11_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL11_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL11_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL11_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL11_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL11_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL11_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.46 CFG\_DATA\_LL11\_LPHDR\_VAL Register (Offset = B8h) [Reset = 00000000h]

CFG\_DATA\_LL11\_LPHDR\_VAL is shown in [Table 22-51](#).

Return to the [Summary Table](#).

**Table 22-51. CFG\_DATA\_LL11\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL11_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.47 CFG\_DATA\_LL11\_THRESHOLD Register (Offset = BCh) [Reset = 0000X0X0h]

CFG\_DATA\_LL11\_THRESHOLD is shown in [Table 22-52](#).

Return to the [Summary Table](#).

**Table 22-52. CFG\_DATA\_LL11\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll11dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL11_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL11_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.48 CFG\_DATA\_LL12 Register (Offset = C0h) [Reset = 00000000h]

CFG\_DATA\_LL12 is shown in [Table 22-53](#).

Return to the [Summary Table](#).

**Table 22-53. CFG\_DATA\_LL12 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL12_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL12_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL12_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL12_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL12_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL12_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL12_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL12_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL12_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL12_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL12_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL12_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL12_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL12_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL12_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.49 CFG\_DATA\_LL12\_LPHDR\_VAL Register (Offset = C4h) [Reset = 00000000h]

CFG\_DATA\_LL12\_LPHDR\_VAL is shown in [Table 22-54](#).

Return to the [Summary Table](#).

**Table 22-54. CFG\_DATA\_LL12\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL12_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.50 CFG\_DATA\_LL12\_THRESHOLD Register (Offset = C8h) [Reset = 0000X0X0h]

CFG\_DATA\_LL12\_THRESHOLD is shown in [Table 22-55](#).

Return to the [Summary Table](#).

**Table 22-55. CFG\_DATA\_LL12\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll12dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL12_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL12_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model



### 22.3.1.51 CFG\_DATA\_LL13 Register (Offset = CCh) [Reset = 00000000h]

CFG\_DATA\_LL13 is shown in [Table 22-56](#).

Return to the [Summary Table](#).

**Table 22-56. CFG\_DATA\_LL13 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL13_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL13_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL13_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL13_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL13_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL13_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL13_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL13_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL13_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL13_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL13_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL13_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL13_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL13_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL13_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.52 CFG\_DATA\_LL13\_LPHDR\_VAL Register (Offset = D0h) [Reset = 00000000h]

CFG\_DATA\_LL13\_LPHDR\_VAL is shown in [Table 22-57](#).

Return to the [Summary Table](#).

**Table 22-57. CFG\_DATA\_LL13\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL13_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.53 CFG\_DATA\_LL13\_THRESHOLD Register (Offset = D4h) [Reset = 0000X0X0h]

CFG\_DATA\_LL13\_THRESHOLD is shown in [Table 22-58](#).

Return to the [Summary Table](#).

**Table 22-58. CFG\_DATA\_LL13\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll13dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL13_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL13_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.54 CFG\_DATA\_LL14 Register (Offset = D8h) [Reset = 00000000h]

CFG\_DATA\_LL14 is shown in [Table 22-59](#).

Return to the [Summary Table](#).

**Table 22-59. CFG\_DATA\_LL14 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL14_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL14_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL14_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL14_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL14_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL14_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL14_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL14_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL14_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL14_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL14_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL14_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL14_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL14_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL14_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.55 CFG\_DATA\_LL14\_LPHDR\_VAL Register (Offset = DCh) [Reset = 00000000h]

CFG\_DATA\_LL14\_LPHDR\_VAL is shown in [Table 22-60](#).

Return to the [Summary Table](#).

**Table 22-60. CFG\_DATA\_LL14\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL14_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.56 CFG\_DATA\_LL14\_THRESHOLD Register (Offset = E0h) [Reset = 0000X0X0h]

CFG\_DATA\_LL14\_THRESHOLD is shown in [Table 22-61](#).

Return to the [Summary Table](#).

**Table 22-61. CFG\_DATA\_LL14\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll14dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL14_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL14_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.57 CFG\_DATA\_LL15 Register (Offset = E4h) [Reset = 00000000h]

CFG\_DATA\_LL15 is shown in [Table 22-62](#).

Return to the [Summary Table](#).

**Table 22-62. CFG\_DATA\_LL15 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL15_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL15_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL15_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL15_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL15_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL15_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL15_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL15_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL15_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL15_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL15_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL15_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL15_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL15_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL15_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.58 CFG\_DATA\_LL15\_LPHDR\_VAL Register (Offset = E8h) [Reset = 00000000h]

CFG\_DATA\_LL15\_LPHDR\_VAL is shown in [Table 22-63](#).

Return to the [Summary Table](#).

**Table 22-63. CFG\_DATA\_LL15\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL15_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB



### 22.3.1.59 CFG\_DATA\_LL15\_THRESHOLD Register (Offset = ECh) [Reset = 0000X0X0h]

CFG\_DATA\_LL15\_THRESHOLD is shown in [Table 22-64](#).

Return to the [Summary Table](#).

**Table 22-64. CFG\_DATA\_LL15\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll15dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL15_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL15_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.60 CFG\_DATA\_LL16 Register (Offset = F0h) [Reset = 00000000h]

CFG\_DATA\_LL16 is shown in [Table 22-65](#).

Return to the [Summary Table](#).

**Table 22-65. CFG\_DATA\_LL16 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL16_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL16_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL16_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL16_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL16_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL16_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL16_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL16_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL16_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL16_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL16_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL16_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL16_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL16_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL16_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.61 CFG\_DATA\_LL16\_LPHDR\_VAL Register (Offset = F4h) [Reset = 00000000h]

CFG\_DATA\_LL16\_LPHDR\_VAL is shown in [Table 22-66](#).

Return to the [Summary Table](#).

**Table 22-66. CFG\_DATA\_LL16\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL16_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.62 CFG\_DATA\_LL16\_THRESHOLD Register (Offset = F8h) [Reset = 0000X0X0h]

CFG\_DATA\_LL16\_THRESHOLD is shown in [Table 22-67](#).

Return to the [Summary Table](#).

**Table 22-67. CFG\_DATA\_LL16\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll16dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL16_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL16_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.63 CFG\_DATA\_LL17 Register (Offset = FCh) [Reset = 00000000h]

CFG\_DATA\_LL17 is shown in [Table 22-68](#).

Return to the [Summary Table](#).

**Table 22-68. CFG\_DATA\_LL17 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL17_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL17_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL17_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL17_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL17_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL17_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL17_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL17_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL17_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL17_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL17_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL17_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL17_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL17_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL17_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.64 CFG\_DATA\_LL17\_LPHDR\_VAL Register (Offset = 100h) [Reset = 00000000h]

CFG\_DATA\_LL17\_LPHDR\_VAL is shown in [Table 22-69](#).

Return to the [Summary Table](#).

**Table 22-69. CFG\_DATA\_LL17\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL17_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.65 CFG\_DATA\_LL17\_THRESHOLD Register (Offset = 104h) [Reset = 0000X0X0h]

CFG\_DATA\_LL17\_THRESHOLD is shown in [Table 22-70](#).

Return to the [Summary Table](#).

**Table 22-70. CFG\_DATA\_LL17\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll17dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL17_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL17_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.66 CFG\_DATA\_LL18 Register (Offset = 108h) [Reset = 00000000h]

CFG\_DATA\_LL18 is shown in [Table 22-71](#).

Return to the [Summary Table](#).

**Table 22-71. CFG\_DATA\_LL18 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL18_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL18_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL18_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL18_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL18_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL18_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL18_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL18_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL18_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL18_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL18_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL18_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL18_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL18_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL18_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid



### 22.3.1.67 CFG\_DATA\_LL18\_LPHDR\_VAL Register (Offset = 10Ch) [Reset = 00000000h]

CFG\_DATA\_LL18\_LPHDR\_VAL is shown in [Table 22-72](#).

Return to the [Summary Table](#).

**Table 22-72. CFG\_DATA\_LL18\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL18_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.68 CFG\_DATA\_LL18\_THRESHOLD Register (Offset = 110h) [Reset = 0000X0X0h]

CFG\_DATA\_LL18\_THRESHOLD is shown in [Table 22-73](#).

Return to the [Summary Table](#).

**Table 22-73. CFG\_DATA\_LL18\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll18dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL18_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL18_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.69 CFG\_DATA\_LL19 Register (Offset = 114h) [Reset = 00000000h]

CFG\_DATA\_LL19 is shown in [Table 22-74](#).

Return to the [Summary Table](#).

**Table 22-74. CFG\_DATA\_LL19 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL19_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL19_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL19_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL19_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL19_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL19_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL19_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL19_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL19_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL19_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL19_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL19_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL19_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL19_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL19_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.70 CFG\_DATA\_LL19\_LPHDR\_VAL Register (Offset = 118h) [Reset = 00000000h]

CFG\_DATA\_LL19\_LPHDR\_VAL is shown in [Table 22-75](#).

Return to the [Summary Table](#).

**Table 22-75. CFG\_DATA\_LL19\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL19_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.71 CFG\_DATA\_LL19\_THRESHOLD Register (Offset = 11Ch) [Reset = 0000X0X0h]

CFG\_DATA\_LL19\_THRESHOLD is shown in [Table 22-76](#).

Return to the [Summary Table](#).

**Table 22-76. CFG\_DATA\_LL19\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll19dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL19_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL19_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.72 CFG\_DATA\_LL20 Register (Offset = 120h) [Reset = 00000000h]

CFG\_DATA\_LL20 is shown in [Table 22-77](#).

Return to the [Summary Table](#).

**Table 22-77. CFG\_DATA\_LL20 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL20_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL20_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL20_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL20_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL20_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL20_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL20_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL20_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL20_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL20_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL20_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL20_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL20_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL20_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL20_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.73 CFG\_DATA\_LL20\_LPHDR\_VAL Register (Offset = 124h) [Reset = 00000000h]

CFG\_DATA\_LL20\_LPHDR\_VAL is shown in [Table 22-78](#).

Return to the [Summary Table](#).

**Table 22-78. CFG\_DATA\_LL20\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL20_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.74 CFG\_DATA\_LL20\_THRESHOLD Register (Offset = 128h) [Reset = 0000X0X0h]

CFG\_DATA\_LL20\_THRESHOLD is shown in [Table 22-79](#).

Return to the [Summary Table](#).

**Table 22-79. CFG\_DATA\_LL20\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll20dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL20_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL20_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model



### 22.3.1.75 CFG\_DATA\_LL21 Register (Offset = 12Ch) [Reset = 00000000h]

CFG\_DATA\_LL21 is shown in [Table 22-80](#).

Return to the [Summary Table](#).

**Table 22-80. CFG\_DATA\_LL21 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL21_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL21_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL21_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL21_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL21_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL21_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL21_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL21_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL21_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL21_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL21_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL21_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL21_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL21_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL21_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.76 CFG\_DATA\_LL21\_LPHDR\_VAL Register (Offset = 130h) [Reset = 00000000h]

CFG\_DATA\_LL21\_LPHDR\_VAL is shown in [Table 22-81](#).

Return to the [Summary Table](#).

**Table 22-81. CFG\_DATA\_LL21\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL21_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.77 CFG\_DATA\_LL21\_THRESHOLD Register (Offset = 134h) [Reset = 0000X0X0h]

CFG\_DATA\_LL21\_THRESHOLD is shown in [Table 22-82](#).

Return to the [Summary Table](#).

**Table 22-82. CFG\_DATA\_LL21\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll21dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL21_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL21_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.78 CFG\_DATA\_LL22 Register (Offset = 138h) [Reset = 00000000h]

CFG\_DATA\_LL22 is shown in [Table 22-83](#).

Return to the [Summary Table](#).

**Table 22-83. CFG\_DATA\_LL22 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL22_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL22_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL22_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL22_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL22_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL22_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL22_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL22_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL22_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL22_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL22_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL22_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL22_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL22_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL22_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.79 CFG\_DATA\_LL22\_LPHDR\_VAL Register (Offset = 13Ch) [Reset = 00000000h]

CFG\_DATA\_LL22\_LPHDR\_VAL is shown in [Table 22-84](#).

Return to the [Summary Table](#).

**Table 22-84. CFG\_DATA\_LL22\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL22_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.80 CFG\_DATA\_LL22\_THRESHOLD Register (Offset = 140h) [Reset = 0000X0X0h]

CFG\_DATA\_LL22\_THRESHOLD is shown in [Table 22-85](#).

Return to the [Summary Table](#).

**Table 22-85. CFG\_DATA\_LL22\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll22dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL22_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL22_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.81 CFG\_DATA\_LL23 Register (Offset = 144h) [Reset = 00000000h]

CFG\_DATA\_LL23 is shown in [Table 22-86](#).

Return to the [Summary Table](#).

**Table 22-86. CFG\_DATA\_LL23 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL23_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL23_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL23_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL23_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL23_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL23_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL23_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL23_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL23_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL23_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL23_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL23_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL23_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL23_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL23_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.82 CFG\_DATA\_LL23\_LPHDR\_VAL Register (Offset = 148h) [Reset = 00000000h]

CFG\_DATA\_LL23\_LPHDR\_VAL is shown in [Table 22-87](#).

Return to the [Summary Table](#).

**Table 22-87. CFG\_DATA\_LL23\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL23_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB



### 22.3.1.83 CFG\_DATA\_LL23\_THRESHOLD Register (Offset = 14Ch) [Reset = 0000X0X0h]

CFG\_DATA\_LL23\_THRESHOLD is shown in [Table 22-88](#).

Return to the [Summary Table](#).

**Table 22-88. CFG\_DATA\_LL23\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll23dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL23_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL23_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.84 CFG\_DATA\_LL24 Register (Offset = 150h) [Reset = 00000000h]

CFG\_DATA\_LL24 is shown in [Table 22-89](#).

Return to the [Summary Table](#).

**Table 22-89. CFG\_DATA\_LL24 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL24_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL24_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL24_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL24_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL24_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL24_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL24_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL24_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL24_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL24_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL24_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL24_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL24_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL24_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL24_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.85 CFG\_DATA\_LL24\_LPHDR\_VAL Register (Offset = 154h) [Reset = 00000000h]

CFG\_DATA\_LL24\_LPHDR\_VAL is shown in [Table 22-90](#).

Return to the [Summary Table](#).

**Table 22-90. CFG\_DATA\_LL24\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL24_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.86 CFG\_DATA\_LL24\_THRESHOLD Register (Offset = 158h) [Reset = 0000X0X0h]

CFG\_DATA\_LL24\_THRESHOLD is shown in [Table 22-91](#).

Return to the [Summary Table](#).

**Table 22-91. CFG\_DATA\_LL24\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll24dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL24_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL24_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.87 CFG\_DATA\_LL25 Register (Offset = 15Ch) [Reset = 00000000h]

CFG\_DATA\_LL25 is shown in [Table 22-92](#).

Return to the [Summary Table](#).

**Table 22-92. CFG\_DATA\_LL25 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL25_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL25_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL25_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL25_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL25_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL25_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL25_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL25_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL25_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL25_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL25_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL25_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL25_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL25_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL25_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.88 CFG\_DATA\_LL25\_LPHDR\_VAL Register (Offset = 160h) [Reset = 00000000h]

CFG\_DATA\_LL25\_LPHDR\_VAL is shown in [Table 22-93](#).

Return to the [Summary Table](#).

**Table 22-93. CFG\_DATA\_LL25\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL25_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.89 CFG\_DATA\_LL25\_THRESHOLD Register (Offset = 164h) [Reset = 0000X0X0h]

CFG\_DATA\_LL25\_THRESHOLD is shown in [Table 22-94](#).

Return to the [Summary Table](#).

**Table 22-94. CFG\_DATA\_LL25\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll25dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL25_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL25_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.90 CFG\_DATA\_LL26 Register (Offset = 168h) [Reset = 00000000h]

CFG\_DATA\_LL26 is shown in [Table 22-95](#).

Return to the [Summary Table](#).

**Table 22-95. CFG\_DATA\_LL26 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL26_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL26_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL26_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL26_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL26_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL26_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL26_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL26_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL26_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL26_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL26_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL26_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL26_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL26_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL26_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid



### 22.3.1.91 CFG\_DATA\_LL26\_LPHDR\_VAL Register (Offset = 16Ch) [Reset = 00000000h]

CFG\_DATA\_LL26\_LPHDR\_VAL is shown in [Table 22-96](#).

Return to the [Summary Table](#).

**Table 22-96. CFG\_DATA\_LL26\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL26_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.92 CFG\_DATA\_LL26\_THRESHOLD Register (Offset = 170h) [Reset = 0000X0X0h]

CFG\_DATA\_LL26\_THRESHOLD is shown in [Table 22-97](#).

Return to the [Summary Table](#).

**Table 22-97. CFG\_DATA\_LL26\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll26dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL26_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL26_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.93 CFG\_DATA\_LL27 Register (Offset = 174h) [Reset = 00000000h]

CFG\_DATA\_LL27 is shown in [Table 22-98](#).

Return to the [Summary Table](#).

**Table 22-98. CFG\_DATA\_LL27 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL27_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL27_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL27_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL27_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL27_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL27_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL27_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL27_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL27_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL27_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL27_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL27_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL27_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL27_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL27_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.94 CFG\_DATA\_LL27\_LPHDR\_VAL Register (Offset = 178h) [Reset = 00000000h]

CFG\_DATA\_LL27\_LPHDR\_VAL is shown in [Table 22-99](#).

Return to the [Summary Table](#).

**Table 22-99. CFG\_DATA\_LL27\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL27_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.95 CFG\_DATA\_LL27\_THRESHOLD Register (Offset = 17Ch) [Reset = 0000X0X0h]

CFG\_DATA\_LL27\_THRESHOLD is shown in [Table 22-100](#).

Return to the [Summary Table](#).

**Table 22-100. CFG\_DATA\_LL27\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll27dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL27_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL27_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.96 CFG\_DATA\_LL28 Register (Offset = 180h) [Reset = 00000000h]

CFG\_DATA\_LL28 is shown in [Table 22-101](#).

Return to the [Summary Table](#).

**Table 22-101. CFG\_DATA\_LL28 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL28_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL28_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL28_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL28_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL28_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL28_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL28_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL28_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL28_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL28_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0_y 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1_y
6-5	LL28_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL28_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL28_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL28_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL28_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.97 CFG\_DATA\_LL28\_LPHDR\_VAL Register (Offset = 184h) [Reset = 00000000h]

CFG\_DATA\_LL28\_LPHDR\_VAL is shown in [Table 22-102](#).

Return to the [Summary Table](#).

**Table 22-102. CFG\_DATA\_LL28\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL28_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.98 CFG\_DATA\_LL28\_THRESHOLD Register (Offset = 188h) [Reset = 0000X0X0h]

CFG\_DATA\_LL28\_THRESHOLD is shown in [Table 22-103](#).

Return to the [Summary Table](#).

**Table 22-103. CFG\_DATA\_LL28\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll28dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL28_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL28_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model



### 22.3.1.99 CFG\_DATA\_LL29 Register (Offset = 18Ch) [Reset = 00000000h]

CFG\_DATA\_LL29 is shown in [Table 22-104](#).

Return to the [Summary Table](#).

**Table 22-104. CFG\_DATA\_LL29 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL29_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL29_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL29_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL29_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL29_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL29_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL29_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL29_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL29_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL29_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL29_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL29_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL29_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL29_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL29_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.100 CFG\_DATA\_LL29\_LPHDR\_VAL Register (Offset = 190h) [Reset = 00000000h]

CFG\_DATA\_LL29\_LPHDR\_VAL is shown in [Table 22-105](#).

Return to the [Summary Table](#).

**Table 22-105. CFG\_DATA\_LL29\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL29_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.101 CFG\_DATA\_LL29\_THRESHOLD Register (Offset = 194h) [Reset = 0000X0X0h]

CFG\_DATA\_LL29\_THRESHOLD is shown in [Table 22-106](#).

Return to the [Summary Table](#).

**Table 22-106. CFG\_DATA\_LL29\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll29dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL29_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL29_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.102 CFG\_DATA\_LL30 Register (Offset = 198h) [Reset = 00000000h]

CFG\_DATA\_LL30 is shown in [Table 22-107](#).

Return to the [Summary Table](#).

**Table 22-107. CFG\_DATA\_LL30 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL30_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL30_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL30_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL30_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL30_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL30_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL30_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL30_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL30_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL30_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL30_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL30_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL30_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL30_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL30_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.103 CFG\_DATA\_LL30\_LPHDR\_VAL Register (Offset = 19Ch) [Reset = 00000000h]

CFG\_DATA\_LL30\_LPHDR\_VAL is shown in [Table 22-108](#).

Return to the [Summary Table](#).

**Table 22-108. CFG\_DATA\_LL30\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL30_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB

### 22.3.1.104 CFG\_DATA\_LL30\_THRESHOLD Register (Offset = 1A0h) [Reset = 0000X0X0h]

CFG\_DATA\_LL30\_THRESHOLD is shown in [Table 22-109](#).

Return to the [Summary Table](#).

**Table 22-109. CFG\_DATA\_LL30\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll30dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL30_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL30_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.105 CFG\_DATA\_LL31 Register (Offset = 1A4h) [Reset = 00000000h]

CFG\_DATA\_LL31 is shown in [Table 22-110](#).

Return to the [Summary Table](#).

**Table 22-110. CFG\_DATA\_LL31 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31	LL31_DATA_WR_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
30	LL31_LONG_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
29	LL31_SHORT_PKT_DELAY_EN	R/W	0h	TI Internal Feature CSI2 only Programming : Use the Packet Delay configured in CFG_DELAY_CONFIG. This is a Debug feature. Not required in Programming model
28	LL31_CRC_EN	R/W	0h	0 : CRC is disabled 1 : This linklist corresponds to ADC Buffer data. Enable the CRC check from ADC Buffer to CBUFF
27	LL31_LPHDR_EN	R/W	0h	CSI2 Programming : 1 : Entry is start of a new CSI-2 packet. Send the LP Payload Header before sending data correspond to this Linklist 0 : Link list is not the start of a Long packet but part of previous packet and hence directly send data LVDS Programming : 1 : Entry is start of a new LVDS Frame 0 : Entry is not the start of the new LVDS Frame
26	LL31_WAITFOR_PKTSENT	R/W	0h	TI Internal Feature Reserved for future debug enhancement 1 : Wait for packet sent signal ack from CSI2 to move forward 0 : Do not wait for packet sent
25-23	LL31_BITPOS_SEL	R/W	0h	TI Internal Feature. Reserved for future use to select which of the 12-bits or 14-bits to be picked up from 16-bit CBUFF unit
22-9	LL31_SIZE	R/W	0h	Configure the Size of the data in terms of the number of samples (not in terms of number of bytes). Sample refers to a 16 bit CBUFF Unit
8	LL31_FMT_IN	R/W	0h	0 : The incoming data sources for this Linklist is aligned to 128-bit 1 : The incoming data sources for this Linklist is aligned to 96-bit
7	LL31_FMT_MAP	R/W	0h	LVDS only : 0 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_0 <sub>y</sub> 1 : Choose CFG_LVDS_MAPPING_LANE <sub>x</sub> _FMT_1 <sub>y</sub>
6-5	LL31_FMT	R/W	0h	Specify the LVDS/CSI2 output format. 00 - 16bit 01 - 14-bit 10 - 12-bit
4-3	LL31_VCNUM	R/W	0h	CSI-2 : Configure the Virtual Channel Number for the Long Packet over which this data is sent
2	LL31_HS	R/W	0h	CSI-2 : 0 : Do not send an Hsync Start packet before sending this data 1 : Send an Hsync Start Packet before sending this data LVDS : 0 : Entry is not the first data of LVDS Frame 1 : Entry is the first data in the LVDS Frame
1	LL31_HE	R/W	0h	CSI-2 : 0 : Do not send an Hsync End packet after sending this data 1 : Send an Hsync End Packet after sending this data LVDS : 0 : Entry is not the last data of LVDS Frame 1 : Entry is the last data in the LVDS Frame
0	LL31_VALID	R/W	0h	0 : Linklist entry is invalid 1 : Linklist entry is valid

### 22.3.1.106 CFG\_DATA\_LL31\_LPHDR\_VAL Register (Offset = 1A8h) [Reset = 00000000h]

CFG\_DATA\_LL31\_LPHDR\_VAL is shown in [Table 22-111](#).

Return to the [Summary Table](#).

**Table 22-111. CFG\_DATA\_LL31\_LPHDR\_VAL Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	LL31_LPHDR_VAL	R/W	0h	CSI-2 Programming : Configure the Long Packet Header to be sent to the Protocol Engine if the LPHDR_EN field is set for the linklist. LVDS Programming : Configure with the static value : 0BBBBBBBB



### 22.3.1.107 CFG\_DATA\_LL31\_THRESHOLD Register (Offset = 1ACh) [Reset = 0000X0X0h]

CFG\_DATA\_LL31\_THRESHOLD is shown in [Table 22-112](#).

Return to the [Summary Table](#).

**Table 22-112. CFG\_DATA\_LL31\_THRESHOLD Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU3	R	0h	
18-16	ll31dman	R/W	0h	If the long Packet Header is enabled, CBUFF can generate a DMA request to trigger the DMA transfer for the new packet 0 : Send a Request on DMA HW Req output line 0 1 : Send a Request on DMA HW Req output line 1 2 : Send a Request on DMA HW Req output line 2 3 : Send a Request on DMA HW Req output line 3 4 : Send a Request on DMA HW Req output line 4 5 : Send a Request on DMA HW Req output line 5 6 : Send a Request on DMA HW Req output line 6 7 : Do not generate dma trigger
15	NU2	R	Bh	
14-8	LL31_WR_THRESHOLD	R/W	0h	Configure the CBUFF FIFO Write threshold over which CBUFF will stall the DMA write to the CBUFF. Static configuration. This can be programmed to fixed value mentioned in the Programming Model
7	NU1	R	Bh	
6-0	LL31_RD_THRESHOLD	R/W	0h	Configure the CBUFF Read threshold to be Reached before sending the data over CSI2/LVDS and start draining the CBUFF FIFO. Static configuration. This can be programmed to fixed value mentioned in the Programming Model

### 22.3.1.108 CFG\_LVDS\_MAPPING\_LANE0\_FMT\_0 Register (Offset = 1B0h) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE0\_FMT\_0 is shown in [Table 22-113](#).

Return to the [Summary Table](#).

**Table 22-113. CFG\_LVDS\_MAPPING\_LANE0\_FMT\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE0_FMT_0_H	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
27-24	CFG_LVDS_MAPPING_LANE0_FMT_0_G	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
23-20	CFG_LVDS_MAPPING_LANE0_FMT_0_F	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
19-16	CFG_LVDS_MAPPING_LANE0_FMT_0_E	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
15-12	CFG_LVDS_MAPPING_LANE0_FMT_0_D	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
11-8	CFG_LVDS_MAPPING_LANE0_FMT_0_C	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
7-4	CFG_LVDS_MAPPING_LANE0_FMT_0_B	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
3-0	CFG_LVDS_MAPPING_LANE0_FMT_0_A	R/W	0h	Lane 0 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 0 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details

### 22.3.1.109 CFG\_LVDS\_MAPPING\_LANE1\_FMT\_0 Register (Offset = 1B4h) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE1\_FMT\_0 is shown in [Table 22-114](#).

Return to the [Summary Table](#).

**Table 22-114. CFG\_LVDS\_MAPPING\_LANE1\_FMT\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE1_FMT_0_H	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
27-24	CFG_LVDS_MAPPING_LANE1_FMT_0_G	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
23-20	CFG_LVDS_MAPPING_LANE1_FMT_0_F	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
19-16	CFG_LVDS_MAPPING_LANE1_FMT_0_E	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
15-12	CFG_LVDS_MAPPING_LANE1_FMT_0_D	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
11-8	CFG_LVDS_MAPPING_LANE1_FMT_0_C	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
7-4	CFG_LVDS_MAPPING_LANE1_FMT_0_B	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details
3-0	CFG_LVDS_MAPPING_LANE1_FMT_0_A	R/W	0h	Lane 1 mapping if Format 0 is selected. Bit [2:0] : 0-7 : Selects the CBUFF unit from the 8 CBUFF units to be sent on Lane 1 Bit 3 0 : Entry is not valid 1 : Entry is valid Please refer to LVDS Mapping Format in Programming model for more details

### 22.3.1.110 CFG\_LVDS\_MAPPING\_LANE2\_FMT\_0 Register (Offset = 1B8h) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE2\_FMT\_0 is shown in [Table 22-115](#).

Return to the [Summary Table](#).

**Table 22-115. CFG\_LVDS\_MAPPING\_LANE2\_FMT\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE2_FMT_0_H	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
27-24	CFG_LVDS_MAPPING_LANE2_FMT_0_G	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
23-20	CFG_LVDS_MAPPING_LANE2_FMT_0_F	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
19-16	CFG_LVDS_MAPPING_LANE2_FMT_0_E	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
15-12	CFG_LVDS_MAPPING_LANE2_FMT_0_D	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
11-8	CFG_LVDS_MAPPING_LANE2_FMT_0_C	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
7-4	CFG_LVDS_MAPPING_LANE2_FMT_0_B	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
3-0	CFG_LVDS_MAPPING_LANE2_FMT_0_A	R/W	0h	Please refer to LVDS Mapping Format section for configuration details

### 22.3.1.111 CFG\_LVDS\_MAPPING\_LANE3\_FMT\_0 Register (Offset = 1BCh) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE3\_FMT\_0 is shown in [Table 22-116](#).

Return to the [Summary Table](#).

**Table 22-116. CFG\_LVDS\_MAPPING\_LANE3\_FMT\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE3_FMT_0_H	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
27-24	CFG_LVDS_MAPPING_LANE3_FMT_0_G	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
23-20	CFG_LVDS_MAPPING_LANE3_FMT_0_F	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
19-16	CFG_LVDS_MAPPING_LANE3_FMT_0_E	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
15-12	CFG_LVDS_MAPPING_LANE3_FMT_0_D	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
11-8	CFG_LVDS_MAPPING_LANE3_FMT_0_C	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
7-4	CFG_LVDS_MAPPING_LANE3_FMT_0_B	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
3-0	CFG_LVDS_MAPPING_LANE3_FMT_0_A	R/W	0h	Please refer to LVDS Mapping Format section for configuration details

### 22.3.1.112 CFG\_LVDS\_MAPPING\_LANE0\_FMT\_1 Register (Offset = 1C0h) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE0\_FMT\_1 is shown in [Table 22-117](#).

Return to the [Summary Table](#).

**Table 22-117. CFG\_LVDS\_MAPPING\_LANE0\_FMT\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE0_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
27-24	CFG_LVDS_MAPPING_LANE0_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
23-20	CFG_LVDS_MAPPING_LANE0_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
19-16	CFG_LVDS_MAPPING_LANE0_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
15-12	CFG_LVDS_MAPPING_LANE0_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
11-8	CFG_LVDS_MAPPING_LANE0_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
7-4	CFG_LVDS_MAPPING_LANE0_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
3-0	CFG_LVDS_MAPPING_LANE0_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for configuration details

### 22.3.1.113 CFG\_LVDS\_MAPPING\_LANE1\_FMT\_1 Register (Offset = 1C4h) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE1\_FMT\_1 is shown in [Table 22-118](#).

Return to the [Summary Table](#).

**Table 22-118. CFG\_LVDS\_MAPPING\_LANE1\_FMT\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE1_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
27-24	CFG_LVDS_MAPPING_LANE1_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
23-20	CFG_LVDS_MAPPING_LANE1_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
19-16	CFG_LVDS_MAPPING_LANE1_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
15-12	CFG_LVDS_MAPPING_LANE1_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
11-8	CFG_LVDS_MAPPING_LANE1_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
7-4	CFG_LVDS_MAPPING_LANE1_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
3-0	CFG_LVDS_MAPPING_LANE1_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for configuration details

### 22.3.1.114 CFG\_LVDS\_MAPPING\_LANE2\_FMT\_1 Register (Offset = 1C8h) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE2\_FMT\_1 is shown in [Table 22-119](#).

Return to the [Summary Table](#).

**Table 22-119. CFG\_LVDS\_MAPPING\_LANE2\_FMT\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE2_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
27-24	CFG_LVDS_MAPPING_LANE2_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
23-20	CFG_LVDS_MAPPING_LANE2_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
19-16	CFG_LVDS_MAPPING_LANE2_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
15-12	CFG_LVDS_MAPPING_LANE2_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
11-8	CFG_LVDS_MAPPING_LANE2_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
7-4	CFG_LVDS_MAPPING_LANE2_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
3-0	CFG_LVDS_MAPPING_LANE2_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for configuration details



### 22.3.1.115 CFG\_LVDS\_MAPPING\_LANE3\_FMT\_1 Register (Offset = 1CCh) [Reset = 00000000h]

CFG\_LVDS\_MAPPING\_LANE3\_FMT\_1 is shown in [Table 22-120](#).

Return to the [Summary Table](#).

**Table 22-120. CFG\_LVDS\_MAPPING\_LANE3\_FMT\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-28	CFG_LVDS_MAPPING_LANE3_FMT_1_H	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
27-24	CFG_LVDS_MAPPING_LANE3_FMT_1_G	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
23-20	CFG_LVDS_MAPPING_LANE3_FMT_1_F	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
19-16	CFG_LVDS_MAPPING_LANE3_FMT_1_E	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
15-12	CFG_LVDS_MAPPING_LANE3_FMT_1_D	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
11-8	CFG_LVDS_MAPPING_LANE3_FMT_1_C	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
7-4	CFG_LVDS_MAPPING_LANE3_FMT_1_B	R/W	0h	Please refer to LVDS Mapping Format section for configuration details
3-0	CFG_LVDS_MAPPING_LANE3_FMT_1_A	R/W	0h	Please refer to LVDS Mapping Format section for configuration details

### 22.3.1.116 CFG\_LVDS\_GEN\_0 Register (Offset = 1D0h) [Reset = 00000000h]

CFG\_LVDS\_GEN\_0 is shown in [Table 22-121](#).

Return to the [Summary Table](#).

**Table 22-121. CFG\_LVDS\_GEN\_0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-30	cpz	R/W	0h	LVDS Clock config. 1 : Clock alignment enabled Others : Internal clock alignment not enabled This needs to be set to 0x1 for correct functionality
29	cblpen	R/W	0h	TI Internal CFG_LASTPULSE_EN
28	cbrcrcen	R/W	0h	LVDS Frame CRC 0 : CRC is not sent at the end of LVDS Frame 1 : CRC is sent at the end of the LVDS Frame
27-24	cfldy	R/W	0h	LVDS FIFO Initial Threshold. This is a Static configuration and could be set to a fixed value as mention in the Programming model
23	cmsbf	R/W	0h	1 : Data is sent out on the LVDS lane MSB first 0 : Data is sent out on the LVDS lane LSB first
22	cpossel	R/W	0h	0 : When a new chirp is starting, align first sample start to negedge of DDR clock. 1 : When a new chirp is starting, align first sample start to posedge of DDR clock (recommended)
21-16	cckdiv	R/W	0h	TI Internal feature. CFG_LVDS_CLK_DIV
15	cclksel1	R/W	0h	TRM Description : 0 : DDR mode clock mux 1 : SDR mode clock mux TI Restricted Description : CFG_LVDS_CLK_SEL1 0-> Use div-by-2 (Q2 path ) 1 -> Used for direct (Q1 path)
14	cclksel	R/W	0h	TI Internal feature. CFG_LVDS_CLK_SEL (between div-by-N and CLK_HSI_DIG) 1 -> CLK_HSI_DIG 0 - through div-by-N (N is programmed in CFG_LVDS_CLK_DIV)
13-12	ckchar	R/W	0h	TI Internal feature. CFG_K_CHAR_SEL
11	ccsmen	R/W	0h	TRM Description : As per alignment TI Restricted Description : 0 : Regular operation 1 : Continuous Streaming Mode Enabled (Not supported internally also in AR16xx)
10	CFG_BIT_CLK_MODE	R/W	0h	Bit Clock Mode 0 : SDR clocking mode 1 : DDR clocking mode
9-8	CFG_LINE_MODE	R/W	0h	TI Internal feature. Reserved.
7	cpkfmt	R/W	0h	TI Internal feature. CFG_PACK_FORMAT: While packing in 12/14 bit whether to use CSI like packing or general packing.
6	cacdsel	R/W	0h	TI Internal feature. CFG_ALL_CHL_READY_DELAY_SEL This bit is added to take of the fast to slow transition in the ADC Buffer. 0 => If the LVDS clock frequency (SDR) is >= 200MHz 1 => If the LVDS clock frequency (SDR) is < 200MHz
5	ctc2en	R/W	0h	TI Internal feature. 0 : Regular operation 1: TC2MODE Enable (Not supported internally also in AR16xx)
4	CFG_8B10B_EN	R/W	0h	TI Internal Feature. Reserved. For Furture enhancement. Not supported in this version 0 : No encoding 1: 8B10B encoding
3	CFG_LVDS_LANE3_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 3 is disbaled 1 : LVDS Lane 3 is enabled

**Table 22-121. CFG\_LVDS\_GEN\_0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
2	CFG_LVDS_LANE2_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 2 is disbaled 1 : LVDS Lane 2 is enabled
1	CFG_LVDS_LANE1_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 1 is disbaled 1 : LVDS Lane 1 is enabled
0	CFG_LVDS_LANE0_EN	R/W	0h	LVDS only programming : 0 : LVDS Lane 0 is disbaled 1 : LVDS Lane 0 is enabled

### 22.3.1.117 CFG\_LVDS\_GEN\_1 Register (Offset = 1D4h) [Reset = 000000XXh]

CFG\_LVDS\_GEN\_1 is shown in [Table 22-122](#).

Return to the [Summary Table](#).

**Table 22-122. CFG\_LVDS\_GEN\_1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-19	NU2	R	0h	RESERVED
18	cgbcen	R/W	0h	TI Internal Feature. 0 : Bit clk is free running 1 : Bit clk is valid only during the valid frame.
17	cfcpol	R/W	0h	TI Internal Feature. 0 : During IDLE, Frame clock will be 0. Start of the valid sample is indicated by the rise edge 1 : During IDLE. Frame clock will be 1. Start of the valid sample is indicated by the fall edge.
16	clfven	R/W	0h	TI Internal feature. Extend the Single Ended Frame Valid When the frame_valid is used as a single ended signal, then make this 1. 0 : Regular Operation. Frame Valid will exactly match with the valid data. 1 : The frame_valid would start early by about 10 lvds_clk (internal) and would extend beyond by 10 lvds_clk (internal) after the end of the frame
15-14	ctpsel3	R/W	0h	TI Internal feature. This is used when Test Pattern Generation Enabled is enabled. 0 :Incremental pattern - For Lane 3
13-12	ctpsel2	R/W	0h	TI Internal feature. This is used when Test Pattern Generation Enabled is enabled. 0 :Incremental pattern - For Lane 2
11-10	ctpsel1	R/W	0h	TI Internal feature. This is used when Test Pattern Generation Enabled is enabled. 0 :Incremental pattern - For Lane 1
9-8	ctpsel0	R/W	0h	TI Internal feature. This is used when Test Pattern Generation Enabled is enabled. 0 :Incremental pattern - For Lane 0
7	NU1	R	Bh	RESERVED
6-4	ctiddly	R/W	0h	TI Internal feature. Configure the skew delay in terms on number of cycles
3	NU3	R	Bh	
2	c3c3l	R/W	0h	LVDS Only Programming: 0 : Regular Operation 1 : Enable 3Ch-3Lane mode in LVDS. Refer to Programming model for more details
1	csdrinv	R/W	0h	TI Internal feature. Configure the clock inversion during SDR mode. 0 : No inversion 1 : Inversion
0	ctpen	R/W	0h	TI Internal feature. 0 : Regular Operation 1 : LVDS Testpattern Enable

### 22.3.1.118 CFG\_LVDS\_GEN\_2 Register (Offset = 1D8h) [Reset = 00000000h]

CFG\_LVDS\_GEN\_2 is shown in [Table 22-123](#).

Return to the [Summary Table](#).

**Table 22-123. CFG\_LVDS\_GEN\_2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_LVDS_GEN_2	R/W	0h	<p>CFG_LVDS_GEN_2[0]: Configure LSB/MSB first for CRC. This feature is supported only when the field CFG_LVDS_GEN_0[28] is set to 1 0 -&gt; The calculated value of 32-bit Ethernet polynomial CRC is swapped and sent out, clear this bit if data is set to LSB first (CFG_LVDS_GEN_0[23]=0) but CRC should be MSB first or vice-versa 1 -&gt; The calculated value of 32-bit Ethernet polynomial CRC is sent out without swapping, set this bit if both data and CRC should have same format (LSB/MSB first) CFG_LVDS_GEN_2[1]: Configure value of frame clock during inter frame period 0 -&gt; Frame clock is held low 1 -&gt; Frame clock is held high CFG_LVDS_GEN_2[2]: Configure frame clock period. This feature is supported only when the field CFG_LVDS_GEN_0[28] is set to 1 0 -&gt; 32-bit CRC is transmitted as single packet with frame clock set to 16h16l (16 high 16 low) configuration 1 -&gt; 32-bit CRC is transmitted as two packets with frame clock set to 8h8l (8 high 8 low) configuration for each packet CFG_LVDS_GEN_2[3]: Configure bit clock during inter frame period 0 -&gt; Bit clock toggles during inter frame period 1 -&gt; Bit clock does not toggle during inter frame period, the value of bit clock is held low This feature is supported when DDR clock is selected (CFG_LVDS_GEN_0[10]=1) and first data sample is driven on posedge of DDR clock (CFG_LVDS_GEN_0[22]=1) CFG_LVDS_GEN_2[4]: Configure CRC inversion. This feature is supported only when the field CFG_LVDS_GEN_0[28] is set to 1 0 -&gt; The calculated value of 32-bit Ethernet polynomial CRC is inverted and sent out 1 -&gt; The calculated value of 32-bit Ethernet polynomial CRC is sent out without inversion CFG_LVDS_GEN_2[5]: Enable/disable the calibration mode, in this mode frame clock will follow data lane[0] 0 -&gt; Calibration mode is disabled 1 -&gt; Calibration mode is enabled</p>

### 22.3.1.119 CFG\_MASK\_REG0 Register (Offset = 1DCh) [Reset = 00000000h]

CFG\_MASK\_REG0 is shown in [Table 22-124](#).

Return to the [Summary Table](#).

**Table 22-124. CFG\_MASK\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG0	R/W	0h	Mask Register field corresponding to STAT_CBUFF_REG0. Refer STAT_CBUFF_REG0 for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occurrence 1 : Event is masked. No interrupt will be generated on occurrence

### 22.3.1.120 CFG\_MASK\_REG1 Register (Offset = 1E0h) [Reset = 00000000h]

CFG\_MASK\_REG1 is shown in [Table 22-125](#).

Return to the [Summary Table](#).

**Table 22-125. CFG\_MASK\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG1	R/W	0h	Mask Register field corresponding to STAT_CBUFF_REG1. Refer STAT_CBUFF_REG1 for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occurrence 1 : Event is masked. No interrupt will be generated on occurrence

### 22.3.1.121 CFG\_MASK\_REG2 Register (Offset = 1E4h) [Reset = 00000000h]

CFG\_MASK\_REG2 is shown in [Table 22-126](#).

Return to the [Summary Table](#).

**Table 22-126. CFG\_MASK\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG2	R/W	0h	Mask Register field corresponding to STAT_LVDS_REG0. Refer STAT_LVDS_REG0 for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occurrence 1 : Event is masked. No interrupt will be generated on occurrence



### 22.3.1.122 CFG\_MASK\_REG3 Register (Offset = 1E8h) [Reset = 00000000h]

CFG\_MASK\_REG3 is shown in [Table 22-127](#).

Return to the [Summary Table](#).

**Table 22-127. CFG\_MASK\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CFG_MASK_REG3	R/W	0h	RESERVED

### 22.3.1.123 STAT\_CBUFF\_REG0 Register (Offset = 1ECh) [Reset = 00000000h]

STAT\_CBUFF\_REG0 is shown in [Table 22-128](#).

Return to the [Summary Table](#).

**Table 22-128. STAT\_CBUFF\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	STAT_CBUFF_REG0_OTHERS	R	0h	Reserved for future enhancement
12	S_FRAME_DONE	R	0h	Indicates that CBUFF has completed sending out data for the current Frame
11	S_CHIRP_DONE	R	0h	Indicates that CBUFF has completed sending out data for the current Chirp
10-6	S_LL_INDEX	R	0h	TI Internal Feature. Debug only. Current Linked list index.
5	S_CSI_PKT_LP_RCVD_STATE	R	0h	TI Internal Feature Indicates that the CSI-2 Packet Received is sent to the CBUFF from the Protocol Engine after Long Data Packet
4	S_CSI_PKT_HE_RCVD_STATE	R	0h	TI Internal Feature Indicates that the CSI-2 Packet Received is sent to the CBUFF from the Protocol Engine after Hsync End Packet
3	S_CSI_PKT_HS_RCVD_STATE	R	0h	TI Internal Feature Indicates that the CSI-2 Packet Received is sent to the CBUFF from the Protocol Engine after Hsync Start Packet
2	S_CSI_PKT_VE_RCVD_STATE	R	0h	TI Internal Feature Indicates that the CSI-2 Packet Received is sent to the CBUFF from the Protocol Engine after Vsync End Packet
1	S_CSI_PKT_VS_RCVD_STATE	R	0h	TI Internal Feature Indicates that the CSI-2 Packet Received is sent to the CBUFF from the Protocol Engine after Vsync Start Packet
0	S_CSI_PKT_RCVD	R	0h	TI Internal Feature Indicates that the CSI-2 Packet Received is sent to the CBUFF from the Protocol Engine

### 22.3.1.124 STAT\_CBUFF\_REG1 Register (Offset = 1F0h) [Reset = 00000000h]

STAT\_CBUFF\_REG1 is shown in [Table 22-129](#).

Return to the [Summary Table](#).

**Table 22-129. STAT\_CBUFF\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-21	S1_UNUSED3	R	0h	
20	S_CBFIFO_READY_IN_FSM	R	0h	TI Internal Feature. Debug only. cbuff-fifo_ready - Keep this masked. Not relevant.
19	S_CBFIFO_EMPTY_IN_FSM	R	0h	TI Internal Feature. Debug only. cbuff-fifo_empty - Keep this masked. Not relevant.
18	S_PKTRCV_ERR	R	0h	TI Internal Feature. Debug only. If the packetReceived arrives at a wrong time. It should NOT be coming while in IDLE state (as no packet was sent before) and in HIBER state (where the next LL group is being evaluated).
17	S_FRAME_ERR	R	0h	Indicates the FrameStart arrived before CBUFF has completed sending out data for all the Chirps programmed
16	S_CHIRP_ERR	R	0h	Indicates tha the chirpAvailable from ADCBuffer arrived before CBUFF has completed sending out the previous Chirp data.
15-12	S1_UNUSED2	R	0h	RESERVED
11	S_CBFIFO_EMPTY	R	0h	TI Internal Feature. Debug only. CBUFF_FIFO Empty Status – Keep this masked, since full and empty will be normal conditions.
10	S_CBFIFO_FULL	R	0h	TI Internal Feature. Debug only. CBUFF_FIFO Full Status – Keep this masked, since full and empty will be normal conditions.
9	S_CBPUSH_ERR	R	0h	TI Internal Feature. Debug only. CBUFF_FIFO_PUSH_ERROR
8	S_CBPOP_ERR	R	0h	TI Internal Feature. Debug only. CBUFF_FIFO_POP_ERROR
7-3	S1_UNUSED1	R	0h	RESERVED
2	S_LCLPUSH_ERR	R	0h	TI Internal Feature. Debug only. LCL_FIFO_PUSH_ERROR
1	S_LCLPOP_ERR	R	0h	TI Internal Feature. Debug only. LCL_FIFO_POP_ERROR
0	S_LCLFSM_ERR	R	0h	TI Internal Feature. Debug only. LCL_FIFO_FSM_ERROR

### 22.3.1.125 STAT\_CBUFF\_REG2 Register (Offset = 1F4h) [Reset = 00000000h]

STAT\_CBUFF\_REG2 is shown in [Table 22-130](#).

Return to the [Summary Table](#).

**Table 22-130. STAT\_CBUFF\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STAT_CBUFF_REG2	R	0h	RESERVED. This does not have corresponding clear or mask

### 22.3.1.126 STAT\_CBUFF\_REG3 Register (Offset = 1F8h) [Reset = 00000000h]

STAT\_CBUFF\_REG3 is shown in [Table 22-131](#).

Return to the [Summary Table](#).

**Table 22-131. STAT\_CBUFF\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STAT_CBUFF_REG3	R	0h	RESERVED. This does not have corresponding clear or mask

### 22.3.1.127 STAT\_LVDS\_REG0 Register (Offset = 1FCh) [Reset = 00000000h]

STAT\_LVDS\_REG0 is shown in [Table 22-132](#).

Return to the [Summary Table](#).

**Table 22-132. STAT\_LVDS\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STAT_LVDS_REG0	R	0h	<p>TI Internal Feature. Debug only. Clr is CLR_LVDS_REG0 and MASK is CFG_MASK_REG2 FSM_STAT_CODE: [3:0] is for Ch0, [7:4] is for Ch1, [11:8] is for Ch2, [15:12] is for ch3 ASYNC_FIFO_STATUS: [19:16] is for Ch0, [23:20] is for Ch1, [27:24] is for Ch2, [32:28] is for ch3 FSM_STATE_CODE : Using this the states can be decoded. ASYNC_FIFO_STATUS: 0 - POP_ERROR 1 - PUSH_ERROR 2- POP_EMPTY 3 - PUSH_FULL. Set the mask for POP_EMPTY and PUSH_FULL. These are normal conditions and will keep happening and need not generate any interrupt</p>

### 22.3.1.128 STAT\_LVDS\_REG1 Register (Offset = 200h) [Reset = 00000000h]

STAT\_LVDS\_REG1 is shown in [Table 22-133](#).

Return to the [Summary Table](#).

**Table 22-133. STAT\_LVDS\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STAT_LVDS_REG1	R	0h	RESERVED

### 22.3.1.129 STAT\_LVDS\_REG2 Register (Offset = 204h) [Reset = 00000000h]

STAT\_LVDS\_REG2 is shown in [Table 22-134](#).

Return to the [Summary Table](#).

**Table 22-134. STAT\_LVDS\_REG2 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STAT_LVDS_REG2	R	0h	RESERVED



### 22.3.1.130 STAT\_LVDS\_REG3 Register (Offset = 208h) [Reset = 00000000h]

STAT\_LVDS\_REG3 is shown in [Table 22-135](#).

Return to the [Summary Table](#).

**Table 22-135. STAT\_LVDS\_REG3 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	STAT_LVDS_REG3	R	0h	RESERVED

### 22.3.1.131 CLR\_CBUFF\_REG0 Register (Offset = 20Ch) [Reset = 00000000h]

CLR\_CBUFF\_REG0 is shown in [Table 22-136](#).

Return to the [Summary Table](#).

**Table 22-136. CLR\_CBUFF\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-13	CLR_CBUFF_REG0_OTHERS	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
12	C_FRAME_DONE	Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
11	C_CHIRP_DONE	Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
10-6	C_LL_INDEX	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
5	C_CSI_PKT_LP_RCVD_STATE	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		

**Table 22-136. CLR\_CBUFF\_REG0 Register Field Descriptions (continued)**

Bit	Field	Type	Reset	Description
4	C_CSI_PKT_HE_RCVD_STATE	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
3	C_CSI_PKT_HS_RCVD_STATE	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
2	C_CSI_PKT_VE_RCVD_STATE	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
1	C_CSI_PKT_VS_RCVD_STATE	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		
0	C_CSI_PKT_RCVD	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG0. Write 0x1 to Clear the field		

### 22.3.1.132 CLR\_CBUFF\_REG1 Register (Offset = 210h) [Reset = 00000000h]

CLR\_CBUFF\_REG1 is shown in [Table 22-137](#).

Return to the [Summary Table](#).

**Table 22-137. CLR\_CBUFF\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CLR_CBUFF_REG1	TI Internal Feature. Clear Register field corresponding to STAT_CBUFF_REG1. Write 0x1 to Clear the field		

### 22.3.1.133 CLR\_LVDS\_REG0 Register (Offset = 214h) [Reset = 00000000h]

CLR\_LVDS\_REG0 is shown in [Table 22-138](#).

Return to the [Summary Table](#).

**Table 22-138. CLR\_LVDS\_REG0 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CLR_LVDS_REG0	TI Internal Feature. Clear Register field corresponding to STAT_LVDS_REG0. Write 0x1 to Clear the field		

### 22.3.1.134 CLR\_LVDS\_REG1 Register (Offset = 218h) [Reset = 00000000h]

CLR\_LVDS\_REG1 is shown in [Table 22-139](#).

Return to the [Summary Table](#).

**Table 22-139. CLR\_LVDS\_REG1 Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CLR_LVDS_REG1	RESERVED		

### 22.3.1.135 STAT\_CBUFF\_ECC\_REG Register (Offset = 21Ch) [Reset = 00000000h]

STAT\_CBUFF\_ECC\_REG is shown in [Table 22-140](#).

Return to the [Summary Table](#).

**Table 22-140. STAT\_CBUFF\_ECC\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	NU2	R	Bh	
9	seccdbe	R	0h	0 : No Double bit error 1 : Indicates a double bit error has occurred
8	seccsbe	R	0h	0 : No Single bit error 1 : Indicates a single bit error has occurred
7-6	NU1	R	0h	
5-0	seccadd	R	0h	6-bit address where the ECC error occurred. It is valid when either seccsbe or seccdbe is set. If none of them is set, then the addr does not mean anything.

### 22.3.1.136 MASK\_CBUFF\_ECC\_REG Register (Offset = 220h) [Reset = 00000000h]

MASK\_CBUFF\_ECC\_REG is shown in [Table 22-141](#).

Return to the [Summary Table](#).

**Table 22-141. MASK\_CBUFF\_ECC\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	NU2	R	Bh	
9	meccdbe	R/W	0h	0 : Double bit error indications are unmasked 1 : Double bit error indications are Masked
8	meccsbe	R/W	0h	0 : Single bit error indications are unmasked 1 : Single bit error indications are Masked
7-0	NU1	R	0h	



### 22.3.1.137 CLR\_CBUFF\_ECC\_REG Register (Offset = 224h) [Reset = 00000000h]

CLR\_CBUFF\_ECC\_REG is shown in [Table 22-142](#).

Return to the [Summary Table](#).

**Table 22-142. CLR\_CBUFF\_ECC\_REG Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-10	NU2	R	Bh	
9	ceccdbe	Clear Register field corresponding to STAT_CBUF_F_ECC. Write 0x1 to Clear the field	0h	
8	ceccsbe	Clear Register field corresponding to STAT_CBUF_F_ECC. Write 0x1 to Clear the field		
7-1	NU1	R		
0	ceccadd	Clear Register field corresponding to STAT_CBUF_F_ECC. Write 0x1 to Clear the field		

### 22.3.1.138 STAT\_SAFETY Register (Offset = 228h) [Reset = 00000000h]

STAT\_SAFETY is shown in [Table 22-143](#).

Return to the [Summary Table](#).

**Table 22-143. STAT\_SAFETY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-9	SAF_UNUSED1	R	0h	RESERVED
8	SAF_CHIRP_ERR	R	0h	Safety Error. Indicates tha the chirpAvailable from ADCBuffer arrived before CBUFF has completed sending out the previous Chirp data.
7-0	SAF_CRC	R	0h	TRM Descrcption : Indicates a CRC error between ADCBuffer and CBUFF. 0 : No Error Non Zero : Error TI Restricted Description : 0 - CRC for col-0 - [15:0], 1 - CRC for col-1 [31:16] 2 - CRC for col-2 [47:32] 3 - CRC for col-3 [63:48] 4 - CRC for col-4 - [79:64] 5 - CRC for col-5 [95:80] 6 - CRC for col-6 [111 :96] 7 - for col-7 [127:112]

### 22.3.1.139 MASK\_SAFETY Register (Offset = 22Ch) [Reset = 00000000h]

MASK\_SAFETY is shown in [Table 22-144](#).

Return to the [Summary Table](#).

**Table 22-144. MASK\_SAFETY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	MASK_SAFETY	R/W	0h	Mask Register field corresponding to STAT_SAFETY. Refer STAT_SAFETY for bitwise mapping. 0 : Event is unmasked and will cause an interrupt on occurrence 1 : Event is masked. No interrupt will be generated on occurrence

### 22.3.1.140 CLR\_SAFETY Register (Offset = 230h) [Reset = 00000000h]

CLR\_SAFETY is shown in [Table 22-145](#).

Return to the [Summary Table](#).

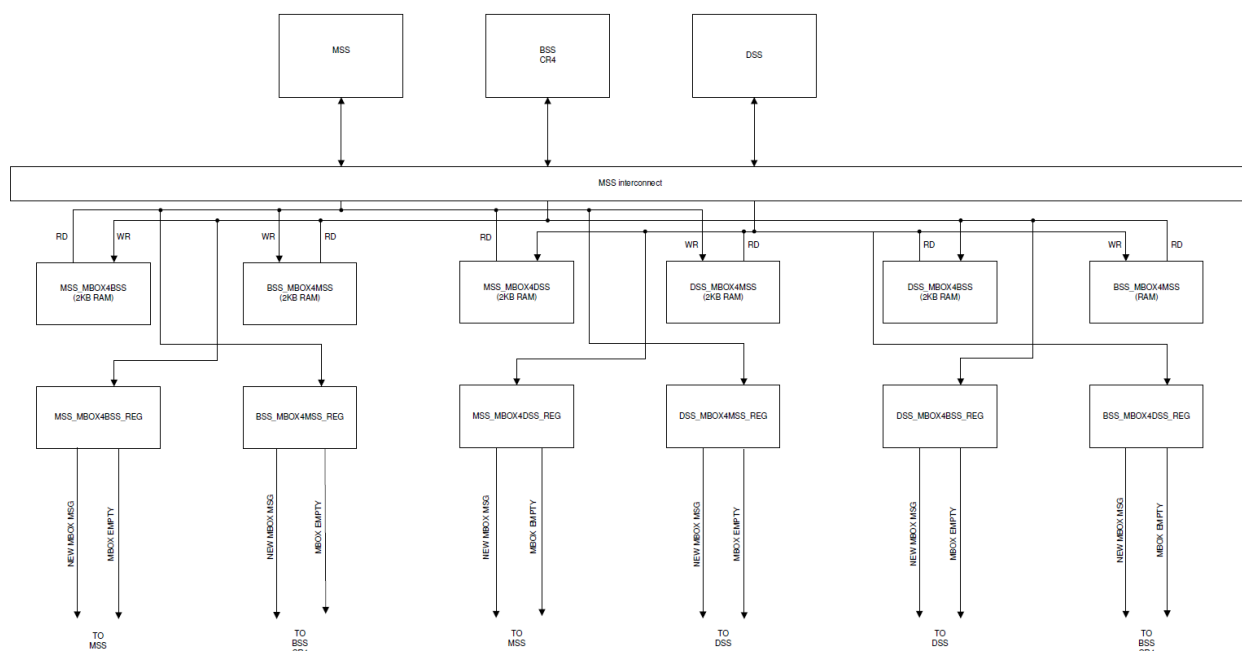
**Table 22-145. CLR\_SAFETY Register Field Descriptions**

Bit	Field	Type	Reset	Description
31-0	CLR_SAFETY	Clear Register field corresponding to STAT_SAFETY. Write 0x1 to Clear the field		



The device provides a mailbox mechanism to asynchronously exchange the messages between any two processors.

Each processor has a mailbox memory space, and registers designated to be used by other processor that wishes to communicate.



**Figure 23-1. Mailbox Block Diagram**

**Table 23-1. Mailbox Allocation for different Sub-System**

Sub-System Name	Size	Start Address (Same across sub-systems)	Mailbox Memory Breakup (* _OFFSET depends on driver implementation)
MSS CR5	4 kB	0xC5000000	RSS_TO_CR5A_MBX 0xC5000000
RSS CR4	2 kB	0xA4030000	CR5A_TO_RSS_MBX: 0xA4030000

### Note

There is an MPU at every Mailbox that can be used to partition the mailbox memory between the Controllers/cores. This gives some flexibility over a fixed allocation scheme.

<b>23.1 Maibox Message Scheme.....</b>	<b>5361</b>
--	-------------

## 23.1 Mailbox Message Scheme

The mailbox architecture is a distributed architecture with the Mailbox memory present in the Receiving processors Subsystem. The following is the processor numbering for AWR2544

**Table 23-2. Mailbox Processor ID**

Processor	ID
MSS_CR5A	0
HSM_CM4	2
RSS_CR4	3

The processor which wishes to send a message to another processor writes the message to the mailbox memory space, then interrupts the receiver processor. The receiver processor acknowledges the interrupt, then reads the message from the mailbox memory space. The receiver informs the sender that the message is read by an interrupt, which is acknowledged back by the sender. The sender must not initiate another message to the same receiver until the previously initiated mailbox interaction with the same receiver is complete.

1. SENDER writes the message in the RECEIVER mailbox.
2. SENDER triggers an interrupt to RECEIVER by writing 1 to <SENDER\_SS>\_CTRL: <SENDER>\_MBOX\_WRITE\_DONE [RECEIVER]. Note: It is writing to its own CTRL space.
3. SENDER polls <RECEIVER\_SS>\_CTRL: <RECEIVER>\_MBOX\_READ\_REQ[SENDER] and keeps writing 1'b1 to <SENDER\_SS>\_CTRL: <SENDER>\_MBOX\_WRITE\_DONE [RECEIVER] until the read value is 1'b1.
4. RECEIVER gets a single interrupt for all interprocessor communication, which is an aggregated interrupt. RECEIVER reads the register <RECEIVER\_SS>\_CTRL::<RECEIVER>\_MBOX\_READ\_REQ and sees bit [SENDER] is 0x1.
5. RECEIVER waits for some 60-70 clock cycles before going to next step.
6. RECEIVER writes to 0x1 to <RECEIVER\_SS>\_CTRL::<RECEIVER>\_MBOX\_READ\_REQ [SENDER] to clear the interrupt.
7. RECEIVER reads the message.
8. RECEIVER writes to 0x1 to <RECEIVER\_SS>\_CTRL::<RECEIVER>\_MBOX\_READ\_DONE\_ACK[SENDER] to generate an acknowledgment interrupt to SENDER. SENDER gets a single interrupt for all interprocessor communication, which is an aggregated ACK interrupt.
9. RECEIVER polls the <SENDER\_SS>\_CTRL: <SENDER>\_MBOX\_READ\_DONE and keeps writing 1'b1 to <RECEIVER\_SS>\_CTRL: <RECEIVER>\_MBOX\_READ\_DONE\_ACK [SENDER] until the read value is 1'b1.
10. SENDER reads the register <SENDER\_SS>\_CTRL: <SENDER>\_MBOX\_READ\_DONE and sees bit [RECEIVER] is 0x1.
11. SENDER waits for some 60-70 clock cycles before going to next step.
12. SENDER writes 0x1 to <SENDER\_SS>\_CTRL: <SENDER>\_MBOX\_READ\_DONE [RECEIVER] to clear the interrupt.

Mailbox message example (message from MSS CR5 C0 to RSS):

1. MSS CR5\_C0 writes the message in the RSS mailbox.
2. MSS triggers an interrupt to RSS by writing 1 to MSS\_CTRL: MSS\_CR5A\_MBOX\_WRITE\_DONE [3]. Note: It is writing to its own CTRL space.
3. MSS CR5\_C0 polls the RSS\_PROC\_CTRL:RSS\_MBOX\_READ\_REQ[0] and keeps writing 1 to MSS\_CTRL: MSS\_CR5A\_MBOX\_WRITE\_DONE [3] until the read value is 1.
4. RSS gets a single interrupt for all interprocessor communication, which is an aggregated interrupt. RSS reads the register RSS\_PROC\_CTRL::RSS\_MBOX\_READ\_REQ and sees bit [0] is 0x1.
5. RSS waits for some 60-70 clock cycles.
6. RSS writes to 0x1 to RSS\_PROC\_CTRL::RSS\_MBOX\_READ\_REQ [0] to clear the interrupt.
7. RSS reads the message.

8. RSS writes to 0x1 to `RSS_PROC_CTRL::RSS_MBOX_READ_DONE_ACK [0]` to generate an acknowledgment interrupt to MSS CR5 C0.
9. RSS polls the `MSS_CTRL::MSS_CR5A_MBOX_READ_DONE[3]` and keeps writing 1 to `RSS_PROC_CTRL::RSS_MBOX_READ_DONE_ACK [0]` until the read value is 1.
10. MSS CR5 C0 gets a single interrupt for all interprocessor communication, which is an aggregated ACK interrupt. MSS CR5 C0 reads the register `MSS_CTRL::MSS_CR5A_MBOX_READ_DONE` and sees bit [3] is 0x1.
11. MSS CR5 C0 waits for some 60-70 clock cycles.
12. MSS CR5 C0 writes 0x1 to `MSS_CTRL::MSS_CR5A_MBOX_READ_DONE [3]` to clear the interrupt.

Every processor is always writing to its own control space.

Each processor has only 2 interrupts (aggregated): `<PROC>MBOX_READ_REQ` and `<PROC>MBOX_READ_ACK` to its interrupt controller.

Scheme ensures the number of mailbox interrupts to a processor is always only 2, regardless of the number of procs in the SoC.



Chapter 24

**Radar Hardware Accelerator 1.5 User Guide**

---



**RadarHardware Accelerator 1.5**

The *RadarHardware Accelerator 1.5 User's Guide* (in three parts) describes the Radar Hardware Accelerator architecture, features, and operation of various blocks and their register descriptions. The purpose is to enable the user to understand the capabilities offered by the Radar Hardware Accelerator and to program it appropriately to achieve the desired functionality.

Radar Hardware Accelerator 1.5 (HWA1.5) is found in devices - AWR2544

This user's guide is divided into three parts. The first part (this document) provides an overview of the overall architecture and features available in the Radar Hardware Accelerator. The main features, such as, windowing, FFT are covered in this part.

The second part of the user's guide covers additional features like DC Estimation/Correction, Interference mitigation and pre-scaling. The second part of the user's guide is optional and can be skipped if the user is interested only in the FFT computation capability.

The third part of user's guide describes the compression feature.

This user's guide (Part 1) is organized as follows: Section 1 covers the introduction and high-level architecture. Section 2, Section 3, and Section 4 describe the state machine, trigger mechanisms, input/output formatting, and general framework for using the accelerator. Section 5 describes the primary computational unit features namely, windowing and FFT.

<b>24.1 Radar Hardware Accelerator 1.5 User Guide – Part 1</b> .....	<b>5364</b>
<b>24.2 RadarHardware Accelerator 1.5 User Guide – Part 2</b> .....	<b>5398</b>
<b>24.3 Radar Hardware Accelerator 1.5 User Guide – Part 3: Compression Engine</b> .....	<b>5419</b>

## 24.1 Radar Hardware Accelerator 1.5 User Guide – Part 1

### RadarHardware Accelerator 1.5

The *RadarHardware Accelerator 1.5 User's Guide* (in three parts) describes the Radar Hardware Accelerator architecture, features, and operation of various blocks and their register descriptions. The purpose is to enable the user to understand the capabilities offered by the Radar Hardware Accelerator and to program it appropriately to achieve the desired functionality.

Radar Hardware Accelerator 1.5 (HWA1.5) is found in devices - AWR2544

This user's guide is divided into three parts. The first part (this document) provides an overview of the overall architecture and features available in the Radar Hardware Accelerator. The main features, such as, windowing, FFT are covered in this part.

The second part of the user's guide covers additional features like DC Estimation/Correction, Interference mitigation and pre-scaling. The second part of the user's guide is optional and can be skipped if the user is interested only in the FFT computation capability.

The third part of user's guide describes the compression feature.

This user's guide (Part 1) is organized as follows: Section 1 covers the introduction and high-level architecture. Section 2, Section 3, and Section 4 describe the state machine, trigger mechanisms, input/output formatting, and general framework for using the accelerator. Section 5 describes the primary computational unit features namely, windowing and FFT.

#### 24.1.1 Radar Hardware Accelerator – Overview

This section provides an overview of the Radar Hardware Accelerator. The section covers the key features of the accelerator and overall architecture.

##### 24.1.1.1 Introduction

The Radar Hardware Accelerator is a hardware IP that enables off-loading the burden of certain frequently used computations in FMCW radar signal processing from the main processor. It is well known that FMCW radar signal processing involves the use of FFT and compression computations to obtain a radar image across the range, velocity, and angle dimensions. Some of the frequently used functions in FMCW radar signal processing can be done within the Radar Hardware Accelerator, while still retaining the flexibility of implementing other proprietary algorithms in the main processor.

This accelerator (HWA 1.5) is optimized for a real-only receiver and has an improved throughput -2 output samples per clock for real signals compared to (HWA 2.0/2.1). It however supports the legacy complex-input and real-input modes of (HWA 2.0/2.1) with single output sample per clock.

##### 24.1.1.2 Key Features

The main features of the Radar Hardware Accelerator are as follows.

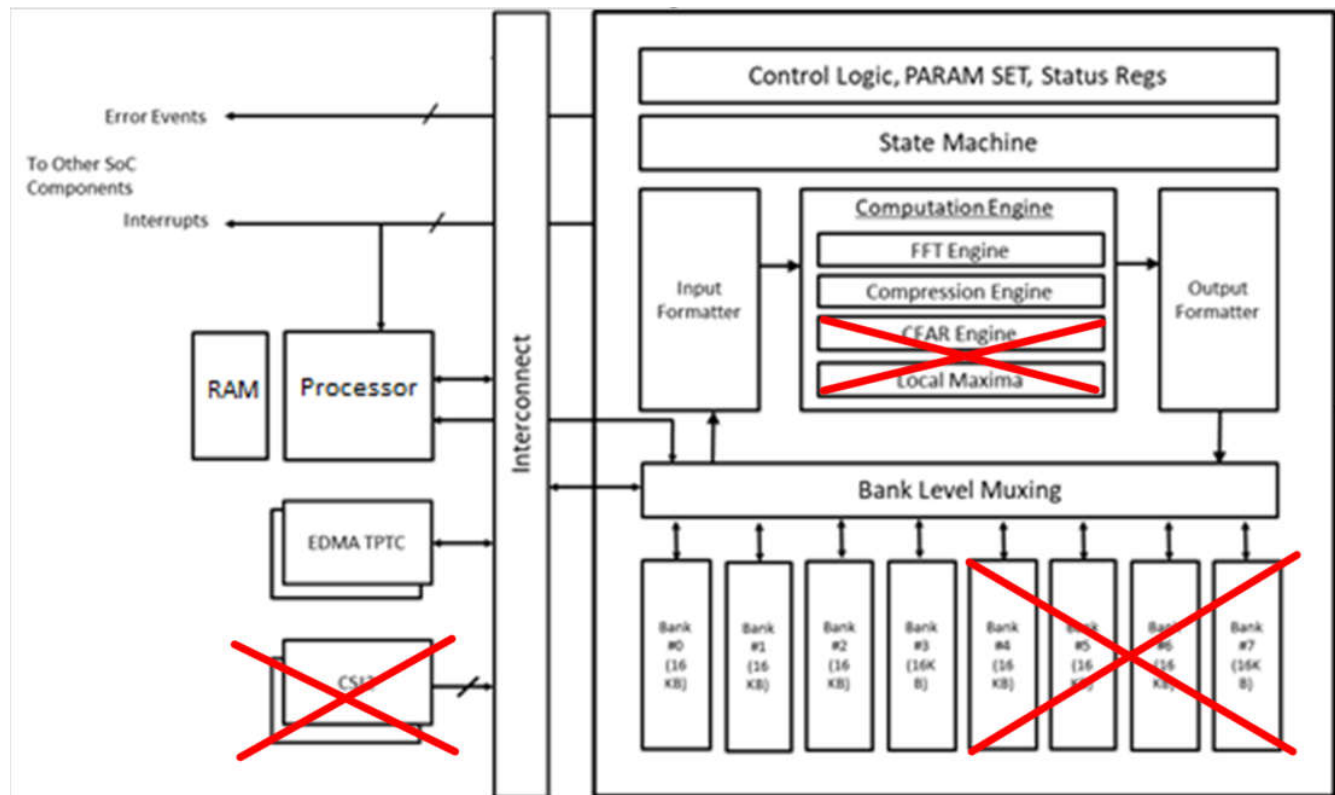
- Fast FFT computation, with programmable FFT sizes (2, 4, 8..., 2048-pt, and 3, 6, 12, ..., 768, 1536-pt) complex-input FFT and (4, 8 ..., 4096-pt and 12, ..., 3072-pt) real-input FFT with improved throughput
- Internal FFT bit width of 24 bits (for each I and Q) for good SQNR performance, with fully programmable butterfly scaling at every stage for user flexibility
- Built-in capabilities for pre-FFT processing – specifically DC estimation and removal, interference localization and mitigation, complex/real multiplication and programmable windowing
- Flexible data flow and data sample arrangement to support efficient chaining of chirp processing operations and transpose accesses as required
- Chaining and looping to sequence a set of accelerator operations one-after-another with minimal intervention from the main processor
- Radar data compression

This user's guide is divided into three parts. The first part covers the high-level architecture and key features such as windowing and FFT. The (optional) second part covers additional features such as DC estimation/correction, Interference mitigation, complex/real multiplication. The third part covers compression and decompression.

### 24.1.1.3 High Level Architecture

The Radar Hardware Accelerator module is loosely coupled to the main processor (eg. ARM Cortex R5). The accelerator is connected to a 64-bit bus that is present in the main processor system, as shown in [Figure 24-1](#).

The Radar Hardware Accelerator module comprises an accelerator engine and four memories, each of 16KB size, which are used to send input data to and pull output data from the accelerator engine. These memories are referred to as *local memories* of the Radar Accelerator (ACCEL\_MEM). For convenience, these local memories are referred to as ACCEL\_MEM0, ACCEL\_MEM1, ... and ACCEL\_MEM3.



**Figure 24-1. Radar Hardware Accelerator**

### 24.1.1.3.1 High-Level Data Flow

The typical data flow is that the DMA module is used to bring samples (for example, FFT input samples) into the local memories of the Radar Hardware Accelerator, so that the main accelerator engine can access and process these samples. Once the accelerator processing is done, the DMA module reads the output samples from the local memories of the Radar Hardware Accelerator and stores them back in the Radar data memory for further processing by the main processor.

The purpose behind the four separate local memories (16KB each) inside the Radar Hardware Accelerator is to enable the *ping-pong* mechanism, for both the input and output, such that the DMA write (and read) operations can happen in parallel to the main computational processing of the accelerator. The presence of multiple memories enables such parallelism. For example, the DMA can be configured to write FFT input samples (ping) into ACCEL\_MEM0 and read FFT output samples (ping) from ACCEL\_MEM2. At the same time, the accelerator engine can be working on FFT input samples (pong) from ACCEL\_MEM1 and writing FFT output samples (pong) into ACCEL\_MEM3. However, both the DMA and the accelerator cannot access the same 16KB memory at the same time. This would lead to an error (refer to the MEM\_ACCESS\_ERR\_STATUS register description in [Table 24-3](#)). As will be explained in later sections, the accelerator engine can perform multiple computational steps one after another autonomously. In each step, the input samples are read from one of the four local memories and the output samples are written into another one of the four local memories.

The Radar Hardware Accelerator operates on a single clock domain and the operating clock frequency is 300 MHz.

The accelerator local memories are 64-bits wide, for example, each of the 16KB banks is implemented as 1024 words of 128 bits each. This allows the DMA to bring data into the accelerator local memories efficiently (up to a maximum throughput of 64 bits per interconnect clock cycle, depending upon the DMA configuration). Two ports for accessing the HWA local memories are available and these ports map the same 64KB into two different address spaces, thus allowing simultaneous efficient read and write DMA transfers.

It is important to note that any of the four local memories can be the *source* of the input samples to the accelerator engine and any of the four local memories can be the *destination* for the output samples from the accelerator engine – with the important restriction that the source and destination memories cannot be the same 16 KB bank. Note also that the accelerator local memories do not necessarily need to be used in ping-pong mode and can instead be used as larger 32KB input and output memories, if the use case requires. The address space for the 16KB memories is contiguous (including a wrap-around from the end of ACCEL\_MEM3 to the start of ACCEL\_MEM0). Therefore, the source as well as destination memory addresses can span beyond 16KB.

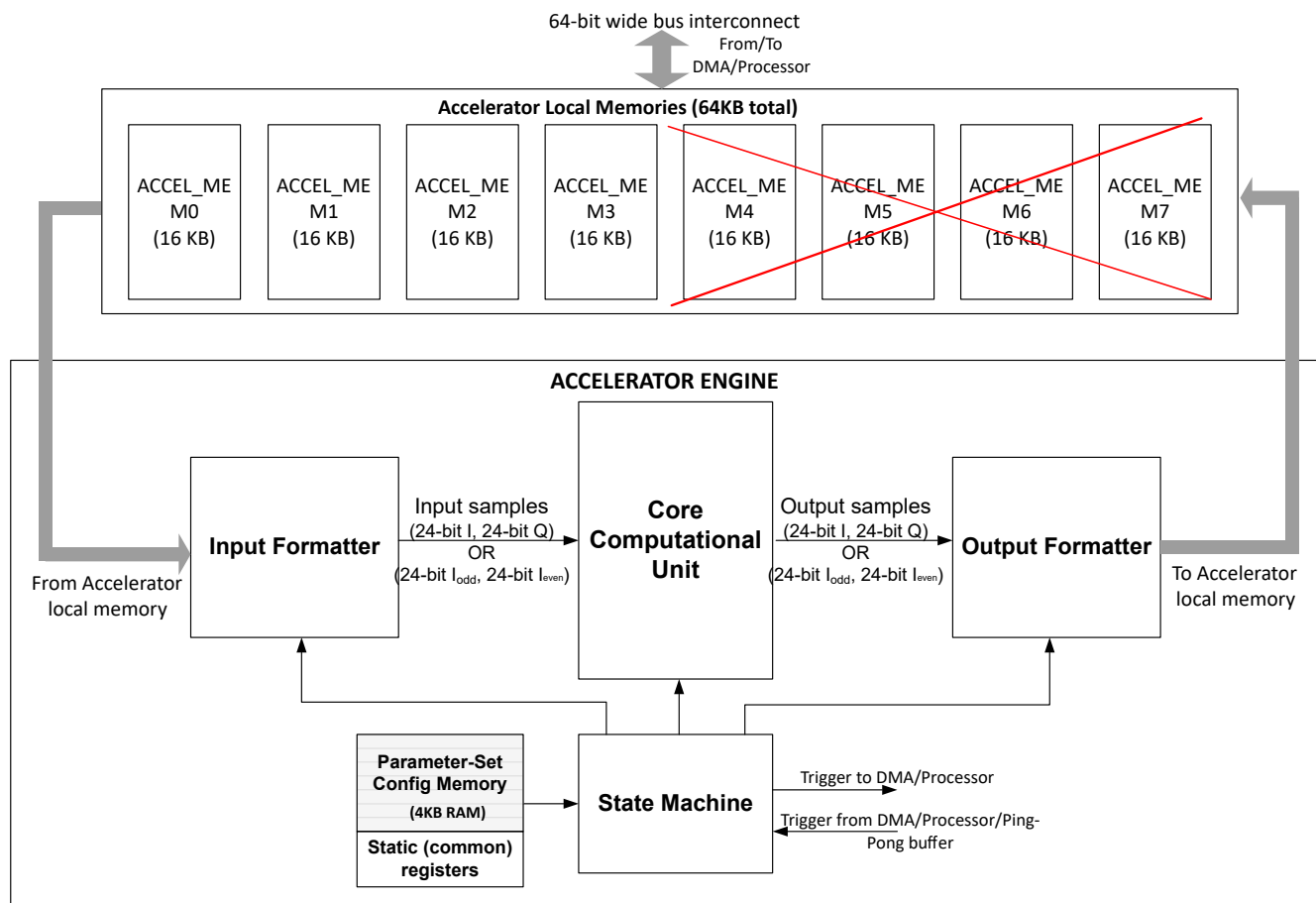
### 24.1.1.3.2 Configuration

The operations of the Radar Hardware Accelerator are configured using registers, which are of two types – *parametersets* and *common* (common for all parameter sets) registers. The purpose of the parameter sets is to enable a complete sequence of various accelerator operations to be preprogrammed (with appropriate source and destination memory addresses and other configurations specified for each operation in that sequence), such that the accelerator can perform them one after the other, with minimal intervention from the main processor.

The parameter-set register configurations are programmed into a separate 4KB *parameter-set configuration memory*. A state machine built into the accelerator handles the loading of one parameter-set configuration at a time and sequences the preprogrammed operations one after another. This process is further explained in later sections of this user's guide.

### 24.1.1.4 Accelerator Engine Block Diagram

As previously mentioned, the Radar Hardware Accelerator module consists of eight local memories of 16KB each (ACCEL\_MEM) and the main accelerator engine. The accelerator engine has the following five components (as shown in [Figure 24-2](#)) – a state machine, input formatter block, output formatter block, core computational unit, and the 4KB parameter-set configuration memory.



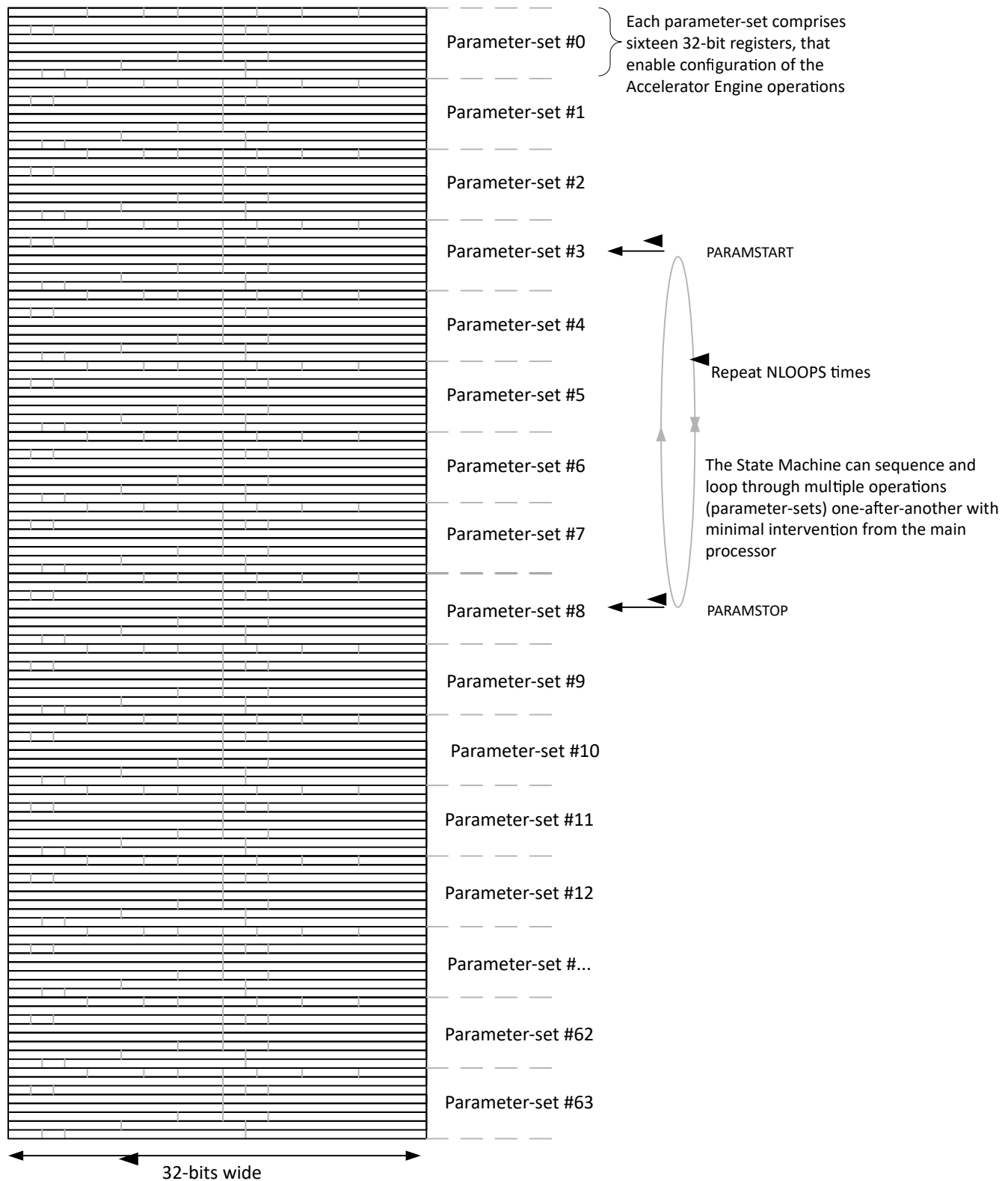
**Figure 24-2. Accelerator Engine Block Diagram. Note that unlike HWA 2.1, HWA 1.5 supports only 4 memory banks of 16KB.**

The purpose of these components is as follows.

- **State machine:** the state machine is responsible for controlling the overall operation of the accelerator – specifically, the starting, looping, stopping, as well as triggering and handshake mechanisms between the accelerator, DMA, and main processor. The state machine is also closely connected to the parameter-set configuration memory and takes care of sequencing and chaining a sequence of multiple accelerator operations as programmed in the parameter-set configuration memory.
- **Input formatter:** the input formatter block is responsible for reading the input samples from any one of the local memories and feeding them into the core computational unit. In this process, this block provides flexible ways of accessing the input samples, in terms of 16-bit versus 32-bit aligned input samples, transpose read-out, flexible scaling, and sign extension to generate internal bit-width of 24 bits, and so on. Lastly the input formatter block provides 24-bit complex samples as input to the core computational unit. The local memory (memories) from which the input formatter reads the input samples is called the *source* memory.
- **Output formatter:** the output formatter block is responsible for writing the output samples from the core computational unit into the local memories. This block also provides flexible ways of formatting the output samples, in terms of 16-bit versus 32-bit aligned output samples, transpose write, flexible scaling from internal bit-width of 24 bits, to 16-bit or 32-bit aligned output samples, sign-extension, and so on. The local memory (memories) to which the output formatter writes the output samples is called the *destination* memory.
- **Core computational unit:** the core computational unit contains the main computational logic for various operations, such as DC correction, windowing and FFT calculations. The unit accepts a streaming input from the input formatter block (at the rate of one input sample per clock cycle), performs computations, and produces a streaming output to the output formatter block (typically at the rate of one output sample per clock cycle), with some initial latency depending on the nature of the computations involved. In HWA 1.5, it

is possible to stream-in two real samples  $I_{\text{odd}}$  and  $I_{\text{even}}$  instead of a complex I-sample and Q-sample. This improves the rate to two sample per clock cycle input. Note that in case of a FFT for purely real-input, the number of unique output samples needed is only half of FFT size. Thus, the output computation time is also bettered

- Parameter-set configuration memory: this is a 4096-byte RAM that is used to preconfigure the sets of parameters (register settings) for a chained sequence of accelerator operations, which can then be executed by the state machine in a loop. This allows the accelerator to perform a preprogrammed sequence of operations in a loop without frequent intervention from the main processor.



**Figure 24-3. Parameter-Set Configuration Memory (4KB)**

The number of parameter sets that can be preconfigured and sequenced (chained) is 64. This means that up to 64 accelerator operations can be chained together and these can then be looped as well, with minimal

intervention from the main processor. For example, operations like DC & Interference correction, FFT and compression can be preconfigured in the parameter-set configuration memory and the state machine can be made to sequence them one after another and run them in a loop for specified number of times. There is a provision available to interrupt the main processor and/or trigger a DMA channel at the end of each parameter set if required. This allows various ways by which the accelerator, DMA, and the main processor can work together to establish a data and processing flow.

As shown in [Figure 24-3](#), each parameter set contains the equivalent of sixteen 32-bit registers, which corresponds to total RAM size of  $64 \times 16 \times 32 \text{ bits} = 4\text{KB}$  for the parameter-set configuration memory.

The layout of the parameter-set register map is provided in [Section 24.1.6](#). Note that the parameter-set RAM must be programmed using 32-bit word writes only (i.e., byte-writes and half-word writes are not supported). The detailed descriptions of the registers are provided in the various sections, as and when the functionality of each component is presented.

Typically, all necessary parameter sets can be pre-configured before triggering execution by the accelerator. If needed, the parameter sets can be modified on-the-fly but without modifying the parameter set being executed at that time. If parameter sets are being overwritten after execution, it is recommended to re-program all the fields for particular set, i.e., avoid partial over-writing.

#### 24.1.1.5 Accelerator Engine Operation

The accelerator engine and the local memories run on a single clock domain. The overall operation of the accelerator can be summarized as follows. The accelerator engine is configured by the main processor through common configuration registers (common for all parameter sets), as well as the parameter-set configuration memory. As explained earlier, the former comprises common register settings for overall control of the accelerator engine, and the latter comprises the 64 parameter-set specific settings which control the functioning of the accelerator for each of its *chained* sequence of operations.

When the accelerator engine is enabled, the state machine kicks off and controls the overall operation of the accelerator, which involves loading the parameter sets one at a time from the parameter-set configuration memory into various internal registers of the accelerator engine and running the accelerator as per the programmed configuration for each parameter set one after another. The entire procedure then repeats in a loop for a programmed number of times (NUMLOOPS described later).

Each parameter set includes various configuration details such as the accelerator mode of operation (FFT, Compression), the source memory address, number of samples, the destination memory address, input formatting, output formatting, trigger mode for controlling the start of computations to ensure proper handshake with the DMA, and so on.

##### 24.1.1.5.1 Data Throughput

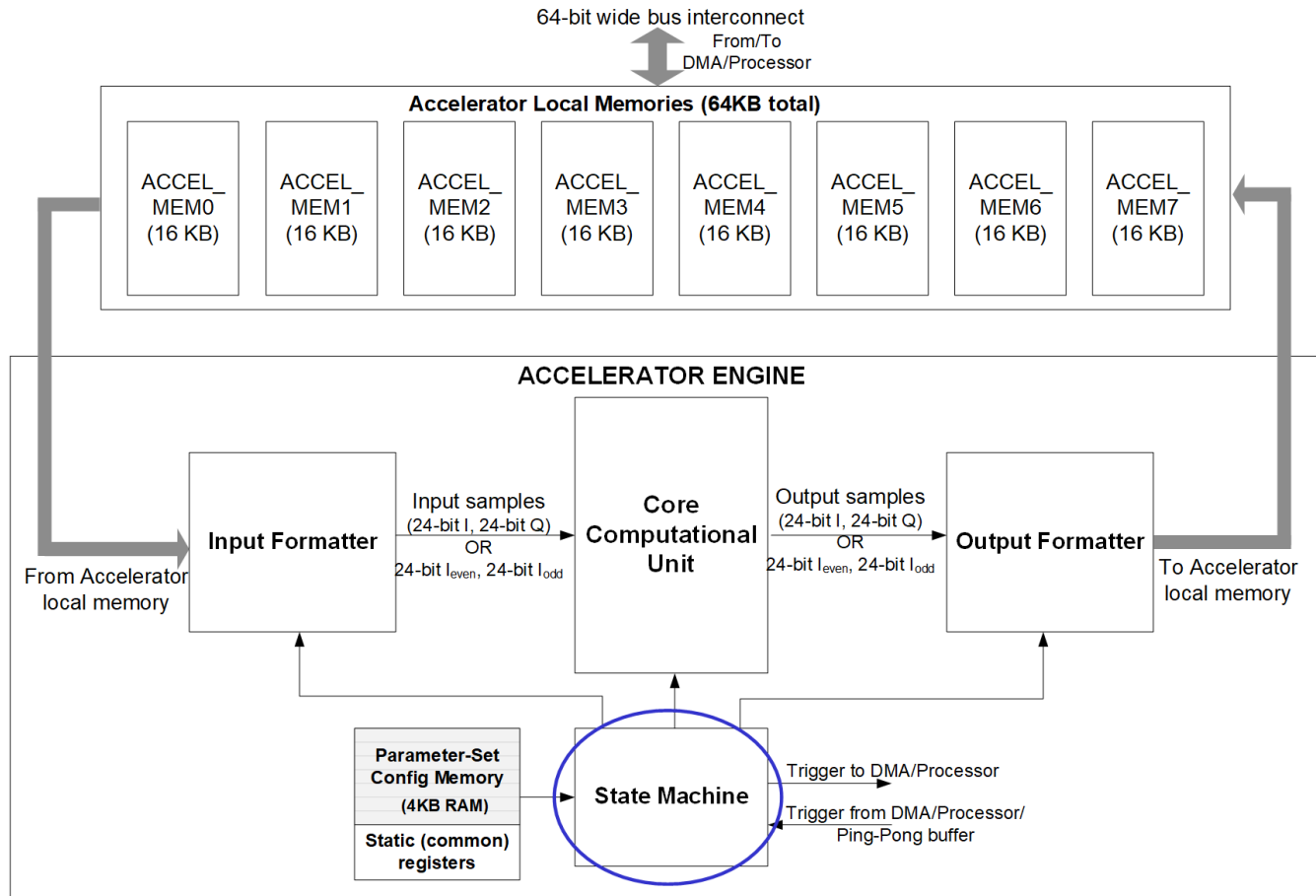
Once the state machine has loaded the registers corresponding to the current parameter set to be executed, the data flow happens as follows: at each clock cycle, one sample from the source memory is read by the input formatter and fed into the core computational unit with appropriate scaling and formatting as configured. The data interface between the input formatter and the core computational unit is a 24-bit complex bus (24-bit for I and 24-bit for Q) which streams either one complex input sample every clock cycle or two real samples per clock cycle. The core computational unit processes this streaming sequence of input samples and in general, produces a streaming output also at one complex-sample every clock cycle or two real-samples, after an initial latency period. Thus, for most operations (DC Estimation, Interference, FFT), in steady state the core computational unit maintains a streaming data rate of one complex-sample per clock cycle or two real-samples per clock cycle. The data interface between the core computational unit and the output formatter is also a 24-bit complex bus (24-bit for I and 24-bit for Q) and the output formatter is responsible for writing into the destination memory, with appropriate scaling and formatting as configured.

The next section provides more details regarding the state machine, including its detailed operation, registers, trigger mechanisms, and so on.



### 24.1.2 Accelerator Engine – State Machine

This section describes the state machine block present in the accelerator engine (see [Figure 24-4](#)). This block, together with the input formatter and output formatter blocks described in the next two sections, provides the overall framework for establishing the data flow and using the accelerator for various computations.



**Figure 24-4. State Machine**

#### 24.1.2.1 State Machine

The statemachine controls the overall functioning of the Radar Hardware Accelerator. The state machine controls the enabling and disabling of the accelerator, as well as supports sequencing an entire set of operations (configured using parameter-set configuration memory), and looping through those operations one after another without needing frequent intervention from the main processor

##### 24.1.2.1.1 State Machine – Operation

The state machine block and the entire accelerator remain in reset and disabled state by default. The state machine (and hence the accelerator in general) is enabled by setting the HWA\_CLK\_EN register bit, followed by writing 111b into the HWA\_EN register.

Notethat a complete list of registers pertaining to the state machine is provided in [Table 24-1](#). Some of the registers are common (common for all parameter sets) registers, whereas some other registers are parameter-set registers, which as explained in the previous section means that they can be uniquely programmed for each of the 64 parameter sets.

For each register, lists whether it is part of the parameter set or not. [Table 24-1](#) also provides a brief description of each register.

When enabled, the state machine steps through (one after another) the parameter sets programmed in the parameter-set configuration memory and executes the computations as per the configuration of each parameter set. The registers `PARAM_START_IDX` and `PARAM_END_IDX` define the starting index and ending index within the 64 parameter sets, so that only those parameter sets between the start and end indices are executed by the accelerator, as shown in [Figure 24-3](#). The state machine also loops through these parameter sets for a total of `NUMLOOPS` times (unless `NUMLOOPS` is programmed as 0 or 4095, in which case the loop does not run or runs infinite times respectively). As an example, if the state machine needs to be configured to run the first four parameter sets in a loop 64 times, then the registers should be programmed as follows: `PARAM_START_IDX = 0`, `PARAM_END_IDX = 3`, and `NUMLOOPS = 64`.

For each parameter set, there is a `TRIGMODE` register, which is used to control when the state machine starts executing the computations for that parameter set. This control is useful, for example, to ensure that the input data is ready in the accelerator local memory (source memory) before the computations are started. Specifically, it is possible to trigger the start of computations after completion of a DMA transfer, or, after a SW interrupt is received. The `TRIGMODE` register setting thus controls when the accelerator operation is triggered for the current parameter set and there are four trigger mechanisms supported as listed in the next subsection. Once triggered, the state machine loads all the registers from the parameter-set configuration memory for the current parameter set into corresponding internal registers of the accelerator and starts the actual computations for that parameter set. After completion of computations of the current parameter set, it moves to the next parameter set. In general, once the parameter set registers are configured and the state machine is enabled, it is recommended to avoid dynamically changing any of the register values. In particular, note that the loading of parameter set (N+1)'s registers happens soon after the completion of parameter set N's execution, and this loading can happen even while the state machine is waiting for the parameter set (N+1)'s trigger to arrive. This means that parameter set configuration registers of parameter set (N+1) cannot be reliably changed while parameter set N is already executing.

After a sequence of operations as programmed in the parameter set(s) for the specified number of loops is complete, the accelerator provides a completion interrupt (insert the correct interrupt name) to the processor. The accelerator can be reconfigured as desired. For reconfiguration, the following procedure must be followed. The accelerator must be disabled by writing 000b to the `HWA_EN` register. Then, a reset must be asserted by writing 111b followed by 000b to the `HWA_RESET` register. The new configurations can now be written in to the accelerator, and then the accelerator can be enabled again by writing 111b to `HWA_EN`. Note that any spare/reserved bits should be kept as 0.

#### 24.1.2.1.2 State Machine – Trigger Mechanisms (Incoming)

As mentioned in the previous subsection, for each parameter set, the start of the computations can be triggered based on specific events. Four trigger mechanisms are supported as follows.

**Immediate trigger (`TRIGMODE = 000b`):** In this case, the state machine does not wait for any trigger and starts the accelerator computations immediately for the current parameter set. This mode is applicable when chaining (sequencing) a set of operations one after another in the accelerator without any need for control handshake or data exchange outside the accelerator (for example, when chaining DC correction and FFT operations) with no need to wait for a trigger in between.

**Wait for processor-based software trigger (`TRIGMODE = 001b`, `TRIGMODE = 111b`):** This is a software-triggered mode that is useful when the main processor(s) must directly control the data flow and start or stop of accelerator computations. In this trigger mode, the state machine waits for a software-based trigger, which involves the main processor(s) setting a separate self-clearing bit in a `FW2HWA_TRIGGER_0` or `FW2HWA_TRIGGER_1` register (single-bit register). The state machine keeps monitoring that register bit and waits as long as the value is zero. When the value becomes 1 (set), the state machine gets triggered to start the accelerator operations for the current parameter set. `FW2HWA_TRIGGER_0` register bit corresponds to `TRIGMODE = 001b` and `FW2HWA_TRG_1` corresponds to `TRIGMODE = 111b`.

**`TRIGMODE = 010b` :** mode is reserved.

**Wait for the DMA-based trigger (TRIGMODE = 011b):** This trigger mode is useful when a DMA transfer completion must be used to trigger the start of the accelerator computations for the current parameter set. The primary purpose of this trigger mode is as follows; when performing first dimension FFT, the DMA is used to bring the FFT input samples from the ADC buffer to the local memory of the accelerator. Upon completion of each DMA transfer, it is useful to automatically trigger the accelerator to perform the FFT.

To achieve this, the state machine of the accelerator has a 32-bit register called the DMA2HWA\_TRIGGER register, where each register bit maps to one of 32 DMA channels that are associated with the accelerator. To use the DMA-based trigger mode, the HWA\_TRIGSRC register in the current parameter set must be programmed to the DMA channel whose completion we wish to monitor. The state machine then monitors the corresponding register bit in the DMA2HWA\_TRIGGER register, and triggers the execution of the current parameter set only when that register bit gets set. For e.g. if HWA\_TRIGSRC is programmed to 5, then the current parameter set will execute only once the register bit #5 gets set in DMA2HWA\_TRIGGER.

The user may utilize the EDMA's linking capability to set the appropriate register bit in DMA2HWA\_TRIGGER. Linking is a programmable feature of the EDMA, where the completion of a DMA transfer can automatically trigger a second DMA transfer. In the present context, the DMA transfer that moves data to the local memory of the accelerator can be linked to a second DMA whose purpose is to write a one-hot signature into DMA2HWA\_TRIGGER to set a specific register bit and trigger the accelerator. Note that there are 32 read-only, one-hot, signature registers (SIG\_DMACH1\_DONE, SIG\_DMACH2\_DONE, and more) that are available. These registers are simply read-only registers which contain hard-coded values (each register is a one-hot signature – 0x0001, 0x0002, 0x0004, 0x0008, and so on). For convenience, these hard-coded 32 read-only signatures can be used, so that the second DMA can simply copy from one of these SIG\_DMACHx\_DONE registers into the DMA2HWA\_TRIGGER register to set the appropriate register bit.

In case of multiple DMA triggers waiting on multiple HWA paramset operation, each time the DMA transfer is completed, the corresponding one-hot signature is copied to DMA2HWA\_TRIGGER (self-clearing bit) register and this value gets internally latched and is serviced only on DMA transfer completion. For example, assume there are two parallel DMA's (DMA-0 and DMA-1) which are about to trigger paramset-0 and paramset-1 respectively. So in this case even if DMA-1 completes transfer first, the HWA will only start execution once DMA-0 is over and also note that DMA2HWA\_TRIGGER register is not overwritten but the values get internally latched as mentioned above.

**Wait for hardware trigger (TRIGMODE = 0b100):** This trigger mode is only applicable in HWA 2.1. In 1.5 this is not supported.

#### 24.1.2.1.3 State Machine – Trigger Mechanisms (Outgoing)

After the accelerator computations for the current parameter set are triggered (using one of the four incoming trigger mechanisms mentioned in the previous subsection), it performs the actual computation operations for that parameter set. These computations typically take several tens or hundreds of clock cycles, depending on the nature of the configuration programmed. Once the accelerator completes its computation operations for the current parameter set, the state machine advances to the next parameter set and repeats the same process. But before advancing to the next parameter set, it can interrupt the main processor and/or trigger a DMA channel. This provision is useful if the main processor is required to read or write registers or memory locations at the end of the current parameter set. Also, this provision is useful for triggering a DMA channel, so that the output of the accelerator can be copied out of the accelerator local memories.

There are two trigger mechanisms provided as follows:

- Interrupt(s) to main processor (CPU\_INTR1\_EN = 1, CPU\_INTR2\_EN = 1): The accelerator interrupts the main processor(s) at the end of completion of computations for the current parameter set, if the register bit CPU\_INTR1\_EN or CPU\_INTR2\_EN is set. Two interrupt signals are available and they are enabled or disabled for each parameter-set by these two register bits. Setting CPU\_INTR1\_EN in a parameter-set enables (interrupt name) interrupt to be generated at the end of that parameter-set. Setting CPU\_INTR2\_EN in a parameter-set enables (interrupt name) interrupt to be generated at the end of that parameter-set.
- Trigger to DMA (DMATRIG\_EN = 1): The accelerator gives a trigger to a DMA channel at the end of completion of computations for the current parameter set, if the register bit DMATRIG\_EN is set. If

DMATRIG\_EN is set, then the particular DMA channel as specified in a separate HWA2DMA\_TRIGDST register (valid values are 0 to 31, for the 32 DMA channels dedicated for the accelerator) is triggered. Thus, it is possible to preconfigure up to 32 DMA channels and trigger the appropriate one at the end of the computations of the current parameter set. The trigger from accelerator to the DMA channels can also be emulated by the processor, by writing to a FW2DMA\_TRIGGER register. This can be used by the processor to kick-start a full/repetitive chain of operations, that are then subsequently managed between the DMA and the accelerator without further processor involvement – for example, the processor writes to the FW2DMA\_TRIGGER register to trigger a DMA channel for the first time, and this kicks off a series of back-to-back data transfers and accelerator computations, with the DMA and accelerator hand-shaking with each other.

#### 24.1.2.1.4 State Machine – Register Descriptions

Table 24-1 lists all the registers of the state machine block. As explained previously, some of the registers are common (common for all parameter sets) registers, whereas some others are *part of each parameter set*. For each register, this distinction is captured as part of the register description in Table 24-1.

**Table 24-1. State Machine Registers**

Register.Field	Width	Parameter Set	Description
HWA_ENABLE.hwa_en	3	No	Enable and Disable Control: This register enables or disables the entire Radar Hardware Accelerator. The reason for a 3-bit register (instead of 1-bit) is to avoid an accidental bit-flip (for example, transient error caused by a neutron strike) from unintentionally turning on the accelerator engine. A value of HWA_EN = 111b enables the Radar Hardware Accelerator and any other value of the register keeps the accelerator engine in disabled state.
HWA_ENABLE.hwa_clk_en	3	No	Clock-gating Control: This register bit controls the enable/disable for the clock of the Radar Accelerator. This register bit can be set to 0 to clock-gate the accelerator when not using the accelerator. Before enabling the accelerator or before configuring the accelerator's registers, this register bit should be set first, so that the clock is available.
HWA_ENABLE.hwa_reset	3	No	Software Reset Control: This register provides software reset control for the Radar Hardware Accelerator. The assertion of these register bits by the main processor will bring the accelerator engine to a known reset state. This is mostly applicable for resetting the accelerator in case of unexpected behavior. Under normal circumstances, it is expected that whenever the accelerator is enabled (from disabled state), it always comes up in a known reset state automatically. The recommended sequence to be followed in case software reset is desired is to write 111b to this register and then a 000b, before the clock is enabled to the accelerator.
PARAM_RAM_LOOP.numloops	12	No	Number of loops: This register controls the number of times the state machine will loop through the parameter sets (from a programmed start index till a programmed end index) and run them. The maximum number of times the loop can be made is run is 4094. A value of 4095 (0xFFFF) programmed in this register should be considered as a special case and it should be interpreted as an infinite loop mode, for example, keep looping and never stop the accelerator engine unless reset by the main processor. A value of zero programmed in this register means that the looping mechanism is disabled. In this case, the accelerator engine can still be used under direct control of the main processor (without the state machine looping provision coming into the picture).
PARAM_RAM_IDX.param_start_idx	10	No	Parameter-set Start Index: These registers are used to control the start and stop index of the parameter set through which the state machine loops through. The state machine starts at the parameter set specified by PARAM_START_IDX and loads each parameter set one after another and runs the accelerator as per that configuration. When the state machine reaches the parameter set specified by PARAM_END_IDX, it loops back to the start index as specified by PARAM_START_IDX.
PARAM_RAM_IDX.param_end_idx	10	No	Parameter-set Stop Index: Refer register description for PARAM_START_IDX

**Table 24-1. State Machine Registers (continued)**

Register.Field	Width	Parameter Set	Description
HWA_ENABLE. hwa_dyn_clk_en	1	No	Dynamic Clock-gating Control: Setting this register bit to '1' enables the capability to clock gate the unused computation engines (i.e., from the two computation engines, namely FFT, Memory Compression) to save power consumption, based on the specific parameter-set being executed.
TRIGMODE	4	Yes	Trigger mode select: 0000b – Immediate trigger 0001b – Software trigger - 1 0010b – Reserved 0011b – DMA-based trigger 0100b – Reserved 0101b – Reserved 0110b – Reserved 0111b – Reserved
FW2HWA_TRIG_0.fw2hwa_trigger_0	1	No	Software trigger bit: This register bit is relevant whenever software triggered mode is used (TRIGMODE = 001b). Whenever this software triggered mode is configured for a parameter set, the state machine keeps monitoring this register bit and waits as long as the value is zero. The main processor software can set this register bit, so that the state machine gets triggered and starts the accelerator operations for that parameter set.
FW2HWA_TRIG_1.fw2hwa_trigger_1	1	No	Software trigger bit: This register bit is similar to FW2HWA_TRIGGER_0, except that this register bit corresponds to TRIGMODE = 111b.
DMA2HWA_TRIG.dma2hwa_trigger	32	No	DMA trigger register: This register is relevant whenever DMA triggered mode is used (for example, TRIGMODE = 011b). Whenever a DMA channel has finished copying input samples into the local memory of the accelerator and wants to trigger the accelerator, the procedure to follow is to use a second linked DMA channel to write a 32-bit one-hot signature into this register to trigger the accelerator. In DMA triggered mode, the state machine keeps monitoring this 32-bit register and waits as long as a specific bit (see DMA2HWA_TRIGSRC) in this register is zero. The second linked DMA channel writes a one-hot signature that sets the specific bit, so that the state machine gets triggered and starts the accelerator operations for that parameter set.
DMA2HWA_TRIGSRC	5	Yes	DMA channel select for DMA completion trigger: This parameter-set register is relevant whenever DMA triggered mode is used (for example, TRIGMODE = 011b). This register selects the bit number in DMA2HWA_TRIGGER for the state machine to monitor to trigger the operation for that parameter set.
CPU_INTR1_EN	1	Yes	Completion interrupt to main processor: This parameter-set register is used to enable/disable interrupt to the main processor upon completion of the accelerator operation for that parameter set. If enabled, the main processor receives an interrupt from the Radar Hardware Accelerator at the end of operations for that parameter set, so that the main processor can take any necessary action. Two interrupts are available, and this register bit enables or disables the first interrupt.
CPU_INTR2_EN	1	Yes	Completion interrupt to main processor: Similar to CPU_INTR1_EN. This register bit enables or disables the second interrupt to the main processor.
PARAM_DONE_SET_STATUS_0 . param_done_set_status_0 PARAM_DONE_SET_STATUS_1. param_done_set_status_1	32	No	Parameter-set done status: These read-only status registers can be used by the main processor to see which parameter sets are complete that led to the interrupt to the main processor. The individual bits in these 32-bit status register indicate which of the 64 parameter sets have completed. These status bits are not automatically cleared, but they can be individually cleared by writing to another set of 32-bit registers (PARAM_DONE_STATUS_CLR)

**Table 24-1. State Machine Registers (continued)**

Register.Field	Width	Parameter Set	Description
PARAM_DONE_STATUS_CLR_0 PARAM_DONE_STATUS_CLR_1	32	No	Refer register description for PARAM_DONE_SET_STATUS
DMATRIG_EN	1	Yes	Completion trigger to DMA: This parameter-set register is used to enable DMA channel trigger upon completion of the accelerator operation for that parameter set. This trigger mechanism enables the accelerator to hand-shake with the DMA so that output data samples are copied out of the accelerator local memory. If enabled, the accelerator triggers a specified DMA channel, so that the output samples can be shipped from the local memory to Radar data memory.
HWA2DMA_TRIGDST	5	Yes	DMA channel select for accelerator completion trigger: This parameter-set register is used to select which of the 32 DMA channels allocated to the accelerator should be triggered upon completion of the accelerator operation for that parameter set. This register is to be used in conjunction with DMATRIG_EN.
FW2DMA_TRIGGER.fw2dma_trigger	32	No	Trigger from processor to DMA: This register can be used by the processor to trigger a DMA channel for the first time, so that a full sequence of repeated operations between the DMA and the accelerator gets kick-started.
PARAMADDR.paramaddr	6	No	Debug register for current parameter-set index: This read-only status register indicates the index of the current parameter set that is under execution. This is useful for debug, where parameter sets can be executed in single-step manner (one-by-one) using SW trigger mode for each of them. In such a debug, this register indicates which parameter set is currently waiting for the SW trigger.
LOOP_CNT.loop_cnt	12	No	Debug register for current loop count: This read-only status register indicates what is the loop count that is presently running. When the state machine is programmed for NUMLOOPS loops, this register shows the current loop count that is running.
TRIGGER_SET_STATUS_0. Trigger_set_status_0	32	No	Debug register for trigger status: This is a read-only status register, which indicates the trigger status of the accelerator, for example, whether a DMA trigger was ever received (refer TRIGMODE). The 32 bits in this register correspond to the 32 DMA trigger bits (refer DMA2HWA_TRIGGER).
TRIGGER_SET_STATUS_1. Trigger_set_status_1	32	No	Debug register for trigger status: This is a read-only status register, which indicates the trigger status of the accelerator, for example, whether software trigger-1 or trigger-2 has even been received.
TRIGGER_SET_IN_CLR_0. Trigger_set_in_clr_0	1	No	Clear trigger status read-only register: This register-bit when set clears the trigger status register TRIGGER_SET_STATUS_0 described above.
TRIGGER_SET_IN_CLR_1 Trigger_set_in_clr_1	1	No	Clear trigger status read-only register: This register-bit when set clears the trigger status register TRIGGER_SET_STATUS_1 described above.

Note that HWA 1.5 does not support context switching unlike HWA 2.0/2.1

The next two sections cover the Input Formatter and Output Formatter blocks, including their detailed operation, registers and usage procedure.



### 24.1.3 Accelerator Engine – Input Formatter

This section describes the input formatter block present in the accelerator engine (see Figure 24-5).

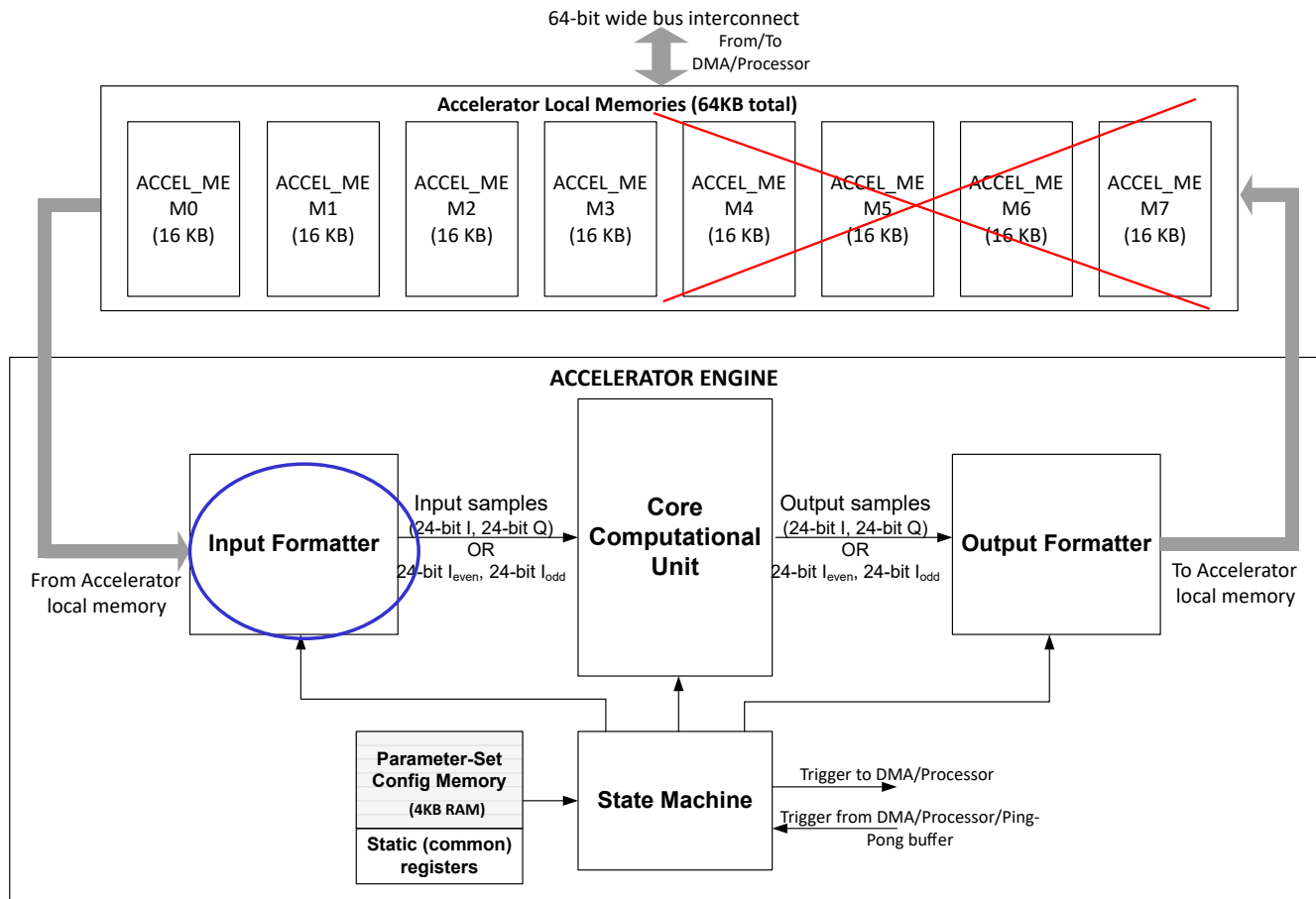


Figure 24-5. Input Formatter

#### 24.1.3.1 Input Formatter

The input formatter is used to access, format, and feed the data from the local memories of the accelerator as 24-bit I and 24-bit Q samples into the core computational unit. The input formatter provides various capabilities to access and format the samples from the local memories – especially, various multidimensional access patterns (for example transpose access), 16-bit or 32-bit aligned word access, scaling using bit-shifts to generate 24-bit wide samples from 16-bit or 32-bit words, real versus complex input, sign extension, conjugation, and more.

##### 24.1.3.1.1 Input Formatter – Operation

The input formatter block is responsible for reading the input samples from the accelerator local memory and feeding them into the core computational unit (see Figure 24-5). The data flow from the input formatter, through the core computational unit, to the output formatter is designed to sustain a steady-state throughput of one complex sample per clock cycle or two real samples per clock cycle. The input formatter thus feeds one sample (24-bit I and 24-bit Q OR 24-bit  $I_{\text{even}}$  and 24-bit  $I_{\text{odd}}$ ) into the core computational unit every clock cycle.

To make the best use of the capabilities of the core computational unit and to allow meaningful chaining of radar signal processing operations with minimal intervention from the processor, the input formatter supports flexibility in how the input samples are accessed from the memory and how they are formatted and fed into the core computational unit.

The memory from which the input formatter picks up the data is referred to as *source memory*. Note that any of the four accelerator local memories can be the source memory. However, as will be described in a subsequent section, there is an important restriction which explains that the source memory cannot be the same as the destination memory (which is the memory to which the output formatter writes the output data).

#### 24.1.3.1.2 Input Formatter – 2D Indexed Addressing for Source Memory Access

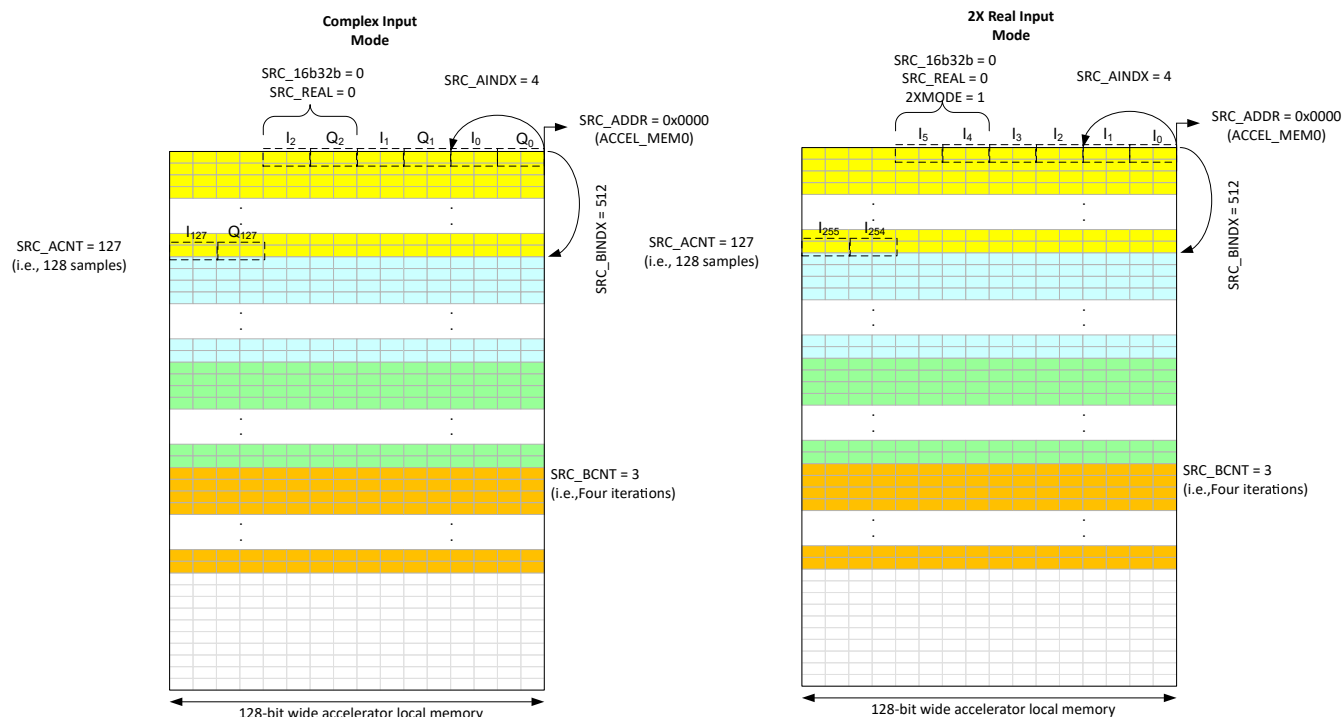
The parameter-set register SRCADDR specifies the start address at which the input samples must be accessed. This register is a byte-address, and a value of 0x00000 corresponds to the first memory location of ACCEL\_MEM0 memory. The SRCADDR register maps to the entire 64KB address space of the four accelerator local memories (4x16KB). Note that the SRCADDR register is a 16-bit register in HWA 1.5, maps to lower 16-bits of 20-bit register in HWA 2.1 maintaining backward compatibility

The input data can be read from the memory as either 16-bit wide samples or 32-bit wide samples. Also, they can be read as real samples or complex samples. These two aspects are configured using register bits SRC16b32b and SRCREAL. See [Table 24-2](#) for a description of these and other registers pertaining to the input formatter block. As an example, if SRC16b32b = 0 and SRCREAL = 0, then the input samples are read from the memory as 16-bit complex samples (16-bit I and 16-bit Q), shown in [Figure 24-6](#). In HWA 1.5, additionally a 2X real input mode is also supported. If 2XMODE = 1 (this register is part of Compute Engine described in Part 2) and SRCREAL = 0, then two consecutive real samples (i.e. odd and even indexed) are fetched simultaneously and streamed in as a pseudo-complex stream. Since the data-path is complex, this packing allows for 2X increase in input rate for real signals. The subsequent compute engine is also adapted to produce 2X real output samples. Note that 2XMODE = 1 and SRCREAL = 1 is invalid configuration. However, SRCREAL = 1 with 2XMODE = 0 is same as legacy 16b input mode of HWA 2.0/2.1. Note that 2X real mode works only with non-interleaved RX ADC data configuration.

Another important feature of the input formatter block is that it supports flexible access pattern to fetch data from the source memory, which makes it convenient when the data corresponding to multiple RX channels are interleaved or when performing multi-dimensional (FFT) processing. This feature is facilitated through the SRCAINDX, SRCACNT, SRCBINDX, and BCNT registers, which are part of each parameter-set configuration.

The register SRCAINDX specifies how many bytes separate successive samples to be fetched from the source memory and the register SRCACNT specifies how many samples need to be fetched per *iteration*. An *iteration* is typically one computational routine, such as one FFT operation. It is possible to perform multiple iterations back-to-back – for example, four FFT operations corresponding to four RX channels. The register SRCBINDX specifies how many bytes separate the start of input samples for successive iterations and BCNT specifies how many iterations to perform back-to-back. These registers can be better understood using the example given in [Figure 24-6](#).





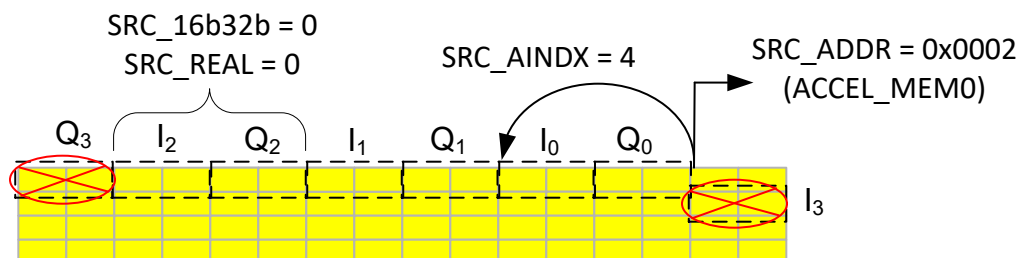
**Figure 24-6. Input Formatter Source Memory Access Pattern (Example)**

In the example shown in [Figure 24-6](#), the input data consists of complex data (16-bit I and 16-bit Q) or pseudo-complex data (16-bit  $I_{\text{even}}$  and 16-bit  $I_{\text{odd}}$ ) that is contiguously present in ACCEL\_MEM0. The data in memory consists of four sets of 128 complex samples each or four sets of 256 real samples (say, corresponding to four RX antennas) and these are shown in four different colors. Because each sample occupies 4 bytes and the samples are contiguously placed in the memory starting at the beginning of ACCEL\_MEM0, values of SRCADDR = 0 and SRCAINDX = 4 are used to fetch these samples.

In each clock cycle, the input formatter fetches one composite sample (I,Q or 2I) from the memory and feeds it into the core computational unit (with appropriate scaling, as described later). Because there are 128 composite samples to be fed for the first iteration (computational routine), a value of SRCACNT = 127 is used. For the second iteration, the samples are fetched starting from a memory location that is SRCBINDX (=128 × 4 = 512) bytes away from SRCADDR.

This process repeats for the programmed number of iterations as per the BCNT register. For example, the value of BCNT = 3 used in this example corresponds to four iterations. Note that the registers shown here are part of parameter-set configuration registers and the four iterations described here can be performed using a single parameter set. Thus, A-dimension is used to run through samples of a given vector, and B-dimension is used to repeat (iterate) the same operation for multiple vectors. Unlike 2.0/2.1 there is no support for a third (C) dimension

An important restriction for complex data, in programming the registers related to source memory access pattern is that the input formatter can only read complex-data from one memory row (128-bit memory location) in a clock cycle. Therefore, if a sample is placed in memory such that the real-part (I value) is at the end of one memory location and the imaginary part (Q value) is at the beginning of the next memory location, then that would be an invalid configuration (see [Figure 24-7](#)). Further, although the accelerator supports byte-addresses, only even values are allowed for SRCADDR, SRCAINDX and SRCBINDX. In 2X Real mode, it is assumed that the pseudo-complex number has two valid real samples in one row.



**Figure 24-7. Invalid Configuration Example**

#### 24.1.3.1.3 Input Formatter – Circular and Shuffled Addressing

The input formatter additionally supports circular addressing in only the A dimension. For the A dimension, the register, `SRCA_CIRCSHIFT` controls the initial offset, while the register, `SRCA_CIRCSHIFTWRAP` defines at which the wrap-around happens.

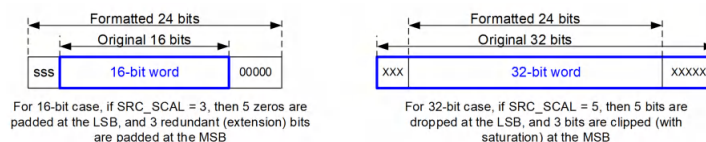
For example, if `SRCA_CIRCSHIFT` is programmed as 7, then the input formatter skips sample indices 0 to 6 and reads samples from the source memory starting directly from index 7 (i.e., the 8<sup>th</sup> sample). Then, for wrap-around, if `SRCA_CIRCSHIFTWRAP` is programmed with a non-zero value, the sample index wraps around (i.e., resets to 0) at  $2^{\text{SRCA\_CIRCSHIFTWRAP}}$ . Continuing the previous example, if `SRCA_CIRCSHIFT` = 7, `SRCA_CIRCSHIFTWRAP` = 9, then the sample indices will be in the following order: 7, 8, 9, 10, 11, ..., 510, 511, 0, 1, 2, 3, 4, 5, 6. In HWA 1.5 there is no support for circular shifting/wrap-around in B-dimension and neither any shuffled LUT based address sequencing in A-dimension

#### 24.1.3.1.4 Input Formatter – Scaling and Formatting

The input formatter allows the input samples read from the source memory to be scaled and formatted before feeding them as 24-bit complex samples into the core computational unit.

Even though the data read from the source memory is initially 16-bits or 32-bits wide (for each  $I_{\text{odd}}$  and  $I_{\text{even}}$ ), the samples expected by the core computational unit are 24-bit complex samples (24-bits each for  $I$  and  $Q$  and 24-bits for each  $I_{\text{odd}}$  and  $I_{\text{even}}$ ). There is a `SRCSICAL` register which provides scaling options using bit-shift to generate 24-bit samples from the original 16- or 32-bit data (see [Figure 24-8](#)).

For the 16-bit case, the 24-bit sample is generated by padding (8-`SRCSICAL`) zeros at the LSB and `SRCSICAL` redundant MSBs. For the 32-bit case, the 24-bit sample is generated by dropping `SRCSICAL` bits at the LSB and clipping (8-`SRCSICAL`) bits at the MSB. Note that the register bit `SRCSIGNED` is used to indicate whether the input samples are signed or unsigned. When this register bit is set, the input samples are treated as signed numbers and hence any extra MSB bits are sign-extended and any clipping of MSB bits takes care of signed saturation. In most cases of interest in part one of this user guide (for example, when performing FFT operation), the input samples would be signed and hence `SRCSIGNED` should be set (i.e., equal to 1).



**Figure 24-8. Input Formatter Data Scaling**

When the input samples are complex (`SRCREAL` = 0) and `2XMODE` = 0, there is a provision to conjugate the input samples. Setting the register bit `SRCCONJ` conjugates the input samples before feeding them to the core computational unit. This feature (together with a corresponding `DSTCONJ` register bit in the output formatter block) enables an IFFT mode from the FFT engine. Note that conjugating the input and output of an FFT block is equivalent to an IFFT function.

There is also provision for swapping  $I$  and  $Q$  samples read from memory. This can be controlled using `SRCIQSWAP` register bit. If `SRCIQSWAP` = 0, then the  $I$  sample is located at the LSB bits, and the  $Q$  sample is located at the MSB bits.

### 24.1.3.1.5 Input Formatter – Register Descriptions

Table 24-2 lists all the registers of the input formatter block.

**Table 24-2. Input Formatter Registers**

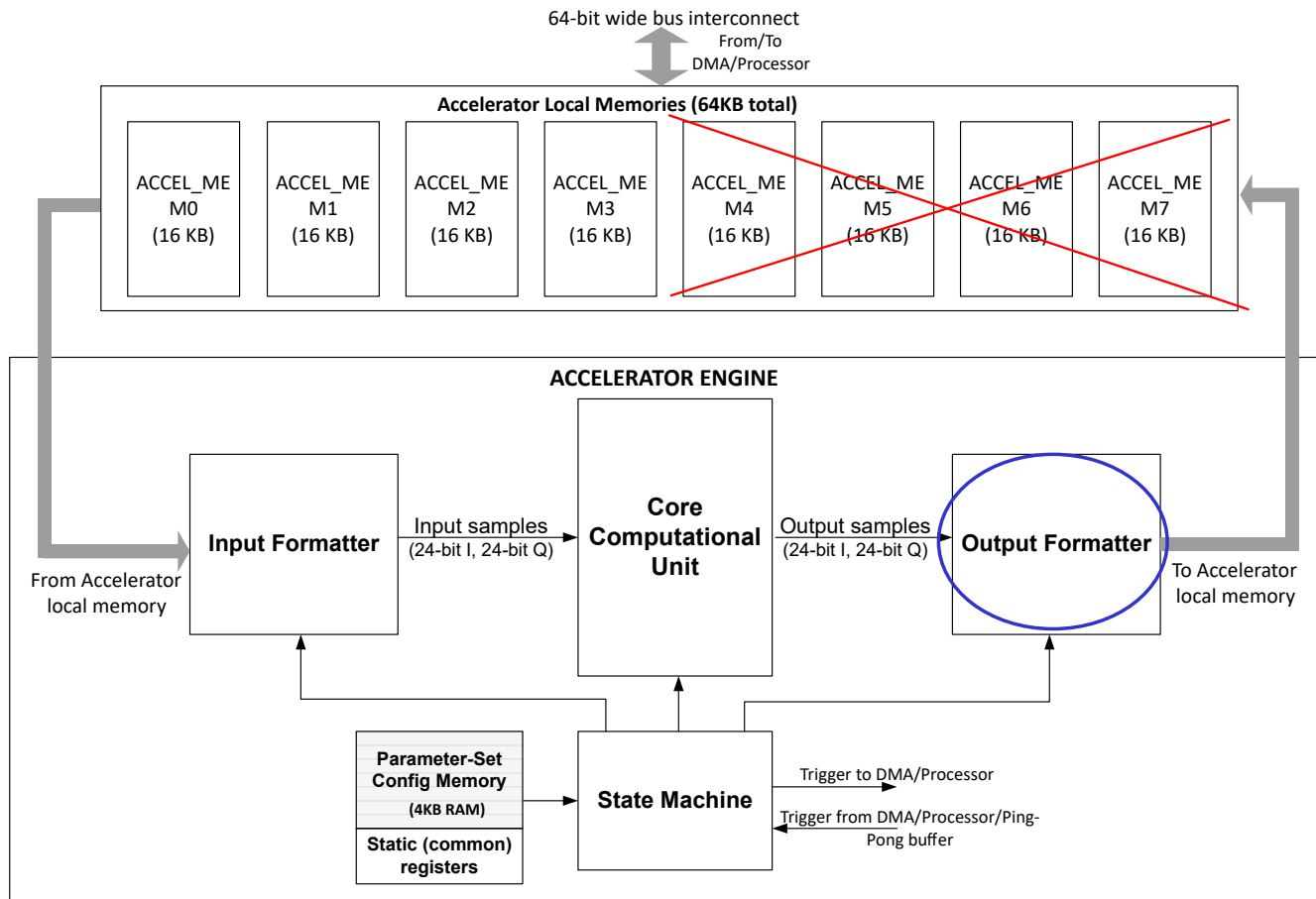
Register	Width	Parameter Set	Description
SRCADDR	16	Yes	Source start address: This register specifies the starting address of the input samples, for example, it specifies the source memory start address from which input samples have to be fetched by the input formatter. This is a byte-address but only even values are valid. This register covers the entire address space of the four local memories ( $4 \times 16\text{KB} = 128\text{KB}$ ). The four accelerator local memories are contiguous in the memory address space and any of them can act as the source memory (as long as the same memory bank is not configured to be used as destination memory at the same time).
SRCACNT	12	Yes	Source sample count: This register specifies the number of samples (minus 1) from the source memory to process for every iteration. The sample count is in number of samples, not number of bytes. For example, the sample count can be specified as 255 (SRCACNT = 0x0FF) in a case where a 256-point complex FFT is required to be performed. For a 256-point real Note however that the sample count register does not always match the FFT size. This can happen when zero-padding of input samples is required. For example, a sample count of 192 could be used with an FFT size of 256, in which case, the input formatter will automatically append 64 zeros.
SRCAINDX	16	Yes	Source sample index increment: This register specifies the number of bytes separating successive samples in the source memory. For example, a value of SRCAINDX = 16 means that successive samples are separated by 16 bytes in memory. Only even values are allowed for this register. The maximum value allowed for this register is 65534.
BCNT	12	Yes	Number of iterations: This register specifies the number of times (minus 1) the processing should be repeated. This register can be used to process the four RX chains back-to-back – for example, a value of BCNT = 3 means that the processing (say first dimension FFT processing) is repeated four times. Note the distinction between the NUMLOOPS register of the state machine block and the BCNT register of the input formatter block. The NUMLOOPS register specifies how many times the state machine loops through all the configured parameter sets (with each time possibly awaiting a trigger), whereas the register BCNT specifies how many times the input formatter and the computational processing of the accelerator is iterated back-to-back for the current parameter set (without any intermediate triggers). Non-zero BCNT should be used only with non-zero ACNT.
SRCBINDX	16	Yes	Source offset per iteration: This register specifies the number of bytes separating the starting address of input samples for successive iterations. For example, when using four iterations to process the four RX chains, this register can be used to specify the offset in the starting address between the successive RX chains. Note the distinction that SRCAINDX specifies the number of bytes separating successive samples for a particular iteration, whereas SRCBINDX specifies the number of bytes separating the starting address of the first sample for successive iterations. Only even values are allowed for this register. The maximum value allowed for this register is 65534.
SRCREAL	1	Yes	Complex or Real Input: This register-bit specifies whether the input samples are real or complex. A value of SRCREAL = 0 implies complex input and a value of SRCREAL = 1 implies real input. When real input is selected, the input formatter block automatically feeds zero for the imaginary part into the core computational unit.
SRCA_CIRCSHIFT	12	Yes	Start index for circular shift in A-dimension. Input Formatter reads samples from the source memory with this start offset to the sample index.

**Table 24-2. Input Formatter Registers (continued)**

Register	Width	Parameter Set	Description
SRCA_CIRCSHIFTWRAP	4	Yes	Circular shift wrap-around point for A-dimension: This register, when set to a non-zero value, specifies the wrap-around point for A-dimension sample counter. If SRC_CIRCSHIFTWRAP3X (A-dimension bit) is set to 0, the A-dimension sample index counter wraps around (i.e., resets to 0) when the counter exceeds $(2^{\text{SRCA\_CIRCSHIFTWRAP}}-1)$ . When that bit is 1, the A-dimension sample index counter wraps around when the counter exceeds $(3 \times 2^{\text{SRCA\_CIRCSHIFTWRAP}}-1)$ .
SRC_CIRCSHIFTWRAP3X	1	Yes	3X enable for circular shift wrap-around: This register is used in conjunction with SRCA_CIRCSHIFTWRAP to specify the wrap-around point when circular shift is used. Refer description of SRCA_CIRCSHIFTWRAP for details.
SRC16b32b	1	Yes	16-bit or 32-bit input word alignment: This register-bit specifies whether the input samples fetched from source memory are to be read as 16-bits or 32-bits wide. A value of SRC16b32b = 0 implies that the input samples are 16-bits wide each (in case of complex input, real and imaginary parts are each 16 bits wide). A value of SRC16b32b = 1 implies that the input samples are 32-bits wide each.
SRCSIGNED	1	Yes	Input sign-extension mode: This register-bit, when set, specifies that the input samples are signed numbers and hence, sign-extension or signed-saturation at the MSB is required when converting 16-bit or 32-bit input words to the 24-bit wide samples to be fed into the core computational unit.
SRCCONJ	1	Yes	Input conjugation: This register-bit specifies whether the input samples should be conjugated before feeding them into the core computational unit. If SRCCONJ is set, then the input samples are conjugated. Setting this register-bit only makes sense if the samples are complex numbers (for example, SRCREAL = 0). This register, together with its counterpart in the output formatter block, enable an IFFT mode for the FFT engine. Note that conjugating the input and output of an FFT block is equivalent to an IFFT function.
SRCSCAL	4	Yes	Input scaling: This register specifies a programmable scaling using bit-shift, when converting the 16-bit or 32-bit wide input data to 24-bit wide samples before feeding into the core computational unit. See Figure 8 and its description for more details regarding this register.
SRCIQSWAP	1	Yes	Swap the I & Q samples drawn from memory. LSB bits drawn from memory is used as I, and the MSB bits are used as Q of input
IP_OP_FORMATTER_CLIP_STAT US. ip_formatter_clip_status	1	No	Read-only register that indicates clip status for input formatter (during scaling).
CLR_CLIP_MISC.clr_clip_status	1	No	Below clip status read-registers will be cleared upon writing to this self-clearing register bit: channel_comb_clip_status, dc_acc_clip_status, dc_est_clip_status, intf_stats_mag_accumulator_clip_status, intf_stats_magdiff_accumulator_clip_status, intf_stats_thresh_mag_clip_status, intf_stats_thresh_magdiff_clip_status, twid_incr_delta_frac_clip_status, ip_formatter_clip_status, op_formatter_clip_status

## 24.1.4 Accelerator Engine – Output Formatter

This section describes the output formatter block present in the accelerator engine (see Figure 24-9).



**Figure 24-9. Output Formatter**

### 24.1.4.1 Output Formatter

The output formatter is used to format and write the data coming out of the core computational unit into the accelerator local memory. Similar to the input formatter block discussed in the previous section, the output formatter block also provides various capabilities to format and write the samples written to the local memory – especially, various two-dimensional access patterns (for example, transpose writes), 16-bit or 32-bit aligned word writes, scaling using bit-shifts to generate 16-bit or 32-bit words from 24-bit wide samples, real versus complex output write, and more.

#### 24.1.4.1.1 Output Formatter – Operation

The output formatter block is responsible for storing the samples coming out of the core computation unit into the accelerator local memory (see Figure 24-9). As mentioned in the previous section, the data flow from the input formatter, through the core computational unit, to the output formatter, is designed to sustain a steady-state throughput of one complex sample per clock cycle. Thus, typically, the output formatter accepts one sample (24-bit I and 24-bit Q) from the core computational unit every clock cycle and writes it to the accelerator local memory. Just like the input formatter, the output formatter also supports lot of flexibility in how the samples are formatted and written into the memory.

The memory into which the output formatter writes the data is referred to as destination memory. Note that any of the four accelerator local memories can be the destination memory, with the important restriction that the source memory cannot be same as the destination memory. In other words, each of the four 16KB memory

banks can either function as source memory, or as destination memory at any time (for example, in any given parameter set).

#### **24.1.4.1.2 Output Formatter – 2-D Indexed Addressing for Destination Memory Access**

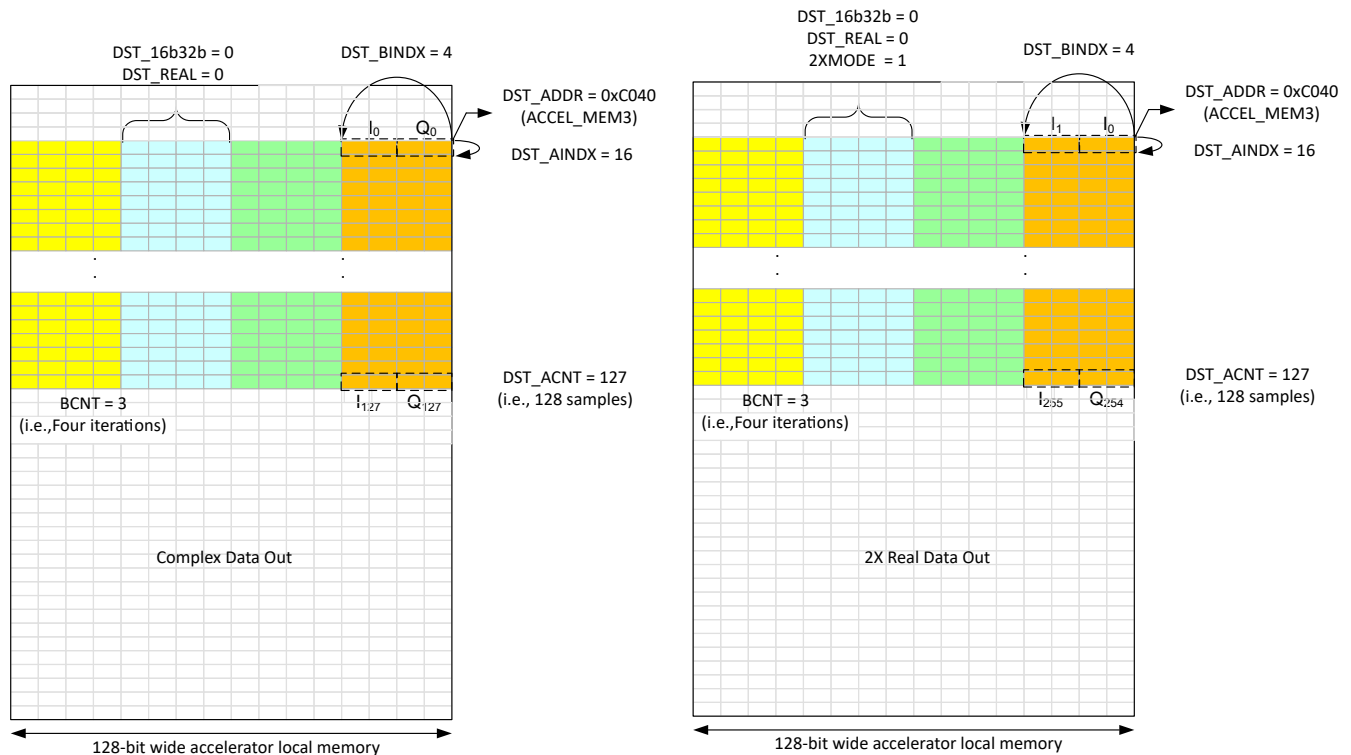
The parameter-set register DSTADDR specifies the start address at which the output samples must be written into the accelerator local memory. Similar to the SRCADDR register of the input formatter, the DSTADDR register of the output formatter is a byte-address and a value of 0x0 corresponds to the first memory location of ACCEL\_MEM0 memory. The DSTADDR register maps to the entire 64KB address space of the four accelerator local memories (each 16KB). As mentioned in the previous paragraph, in a given parameter set, SRCADDR and DSTADDR cannot be configured such that the input samples being fetched and the output samples being written out are accessing the same memory bank.

Even though the core computational unit produces a 24-bit complex output stream, this output data can be written to the memory as either 16-bit wide samples or 32-bit wide samples. Also, they can be written out as complex samples or real samples. These two aspects are configured using register bits DST16b32b and DSTREAL. See [Table 24-3](#) for a description of these and other registers pertaining to the output formatter block. As an example, if DST16b32b = 0 and DSTREAL = 0, then the output samples are written to the memory as 16-bit complex samples (16-bit I and 16-bit Q) or pseudo-complex samples (16-bit I<sub>odd</sub> and 16-bit I<sub>even</sub>) with 2XMODE = 1, shown in [Figure 24-10](#).

Similar to the input formatter block, the output formatter block also supports flexible patterns to write multidimensional data to the destination memory and this makes it convenient when the data corresponding to multiple RX channels must be interleaved, or when performing multidimensional (FFT) processing. This feature is facilitated through the DSTAINDX, DSTACNT, DSTBINDX, and BCNT registers, which are part of each parameter-set configuration.

The register DSTAINDX specifies how many bytes separate successive samples to be written to the destination memory and the register DSTACNT specifies how many samples must be written per iteration. Note that DSTACNT can be different from SRCACNT – this is useful when only a subset of the output samples need to be stored in the output memory (for example, if some FFT output bins must be discarded). The register DSTBINDX specifies how many bytes separate the start of output samples for successive iterations and BCNT specifies the number of iterations. The BCNT register is common for input formatter and output formatter. These registers can be better understood using the example given in [Figure 24-10](#).





**Figure 24-10. Output Formatter Destination Memory Access Pattern (Example)**

In the example shown in [Figure 24-10](#), the output data for example consists of complex data (16-bit I and 16-bit Q) that is written to ACCEL\_MEM3. The output data consists of four sets of 128 samples each (say, corresponding to complex FFT output of four RX antennas) and these are shown in four different colors. The output data could also be 2X real data as shown. Each sample occupies 4 bytes and the samples are written to the output memory at a specific start address inside ACCEL\_MEM3, as shown in [Figure 24-10](#). The samples for the four RX antennas are written to the memory in an interleaved manner. Thus, for this example, a value of DSTADDR = 0xC040, DSTAINDX = 16, DSTACNT = 127, and DSTBINDX = 4 are used. The register BCNT (common for input formatter and output formatter) is configured with a value of 3, corresponding to the four iterations required (for the four RX antennas). In steady state, for each clock cycle, the output formatter accepts one complex sample from the core computational unit and writes it into the memory as per the 2-D indexed addressing pattern programmed.

The register DSTACNT, which corresponds to the number of samples written to the destination memory for each iteration does not need to be equal to SRCACNT. This is useful in cases where some of the complex output samples (for example, some FFT bins at the end) can be dropped and do not need to be written into the destination memory. Another register, DST\_SKIP\_INIT is also available, which can be used to skip some samples in the beginning as well. The number of samples written to the destination memory for each iteration is equal to (DSTACNT + 1) - DST\_SKIP\_INIT.

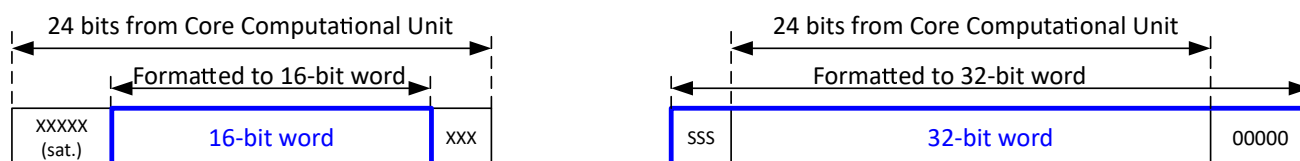
Notethat when performing FFT operations, internally the core computational unit sends out FFT output data in bit-reversed addressing order, but this is automatically handled in the output formatter, such that when the FFT output samples are written into the destination memory, they are written out in the correct normal order. Therefore, no special procedure is required on the part of the main processor to read the FFT output samples in the right sequence.

Similar to the input formatter, the output formatter can write data into only one memory row (128-bit memory location) in a clock cycle. Therefore DSTADDR, DSTAINDX, and DSTBINDX should be programmed such that no sample needs to be partially written in one memory row and the next (e.g. if DST16b32b=1). Also, these address parameters should be restricted to even values.

### 24.1.4.1.3 Output Formatter – Scaling and Formatting

The output formatter allows the 24-bit output samples from the core computational unit to be scaled and formatted before writing them to the destination memory as 16-bit or 32-bit words. There is a DSTSCAL register which provides scaling options using bit-shift, to take the 24-bit samples and convert them to 16-bit or 32-bit data.

For the 16-bit case (Figure 24-11), the 24-bit sample (24-bits for each I and Q) is converted to 16-bit word by dropping DSTSCAL bits at the LSB and by clipping with saturation (8-DSTSCAL) bits at the MSB. For the 32-bit case, the 24-bit sample is padded with DSTSCAL extra bits at the MSB and with (8- DSTSCAL) extra zeros at the LSB. Note that the register bit DSTSIGNED is used to indicate whether the output samples are signed or unsigned. When this register bit is set, the output samples are treated as signed numbers and therefore any extra MSB bits are sign-extended and any clipping of MSB bits handles signed saturation. In most cases of interest in part one of this user's guide (for example, when performing FFT operation), the output samples would be signed and therefore DSTSIGNED should be set (for example, equal to 1). However, if the log-magnitude operation in the core computational unit is enabled, then the output samples are unsigned and therefore DSTSIGNED is cleared (for example, equal to zero).



For 16-bit case, if DST\_SCAL = 3, then 3 bits are dropped at the LSB, and 5 bits are clipped (saturated) at the MSB

For 32-bit case, if DST\_SCAL = 3, then 5 zeros are padded at the LSB, and 3 bits are extended at the MSB

**Figure 24-11. Output Formatter Data Scaling**

When the output samples are complex (for example, DSTREAL = 0), there is a provision to conjugate the output samples. Setting the register bit DSTCONJ conjugates the output samples before writing them to the destination memory. This feature (together with a corresponding SRCCONJ register bit in the input formatter block) enables an IFFT mode from the FFT engine.

In addition, there is provision for swapping I and Q samples written into the destination memory. This is controlled using DSTIQSWAP register bit.

### 24.1.4.1.4 Output Formatter – Register Descriptions

Table 24-3 lists all the registers of the output formatter block.

**Table 24-3. Output Formatter Registers**

Register	Width	Parameter Set	Description
DSTADDR	16	Yes	Destination start address: This register specifies the starting address of the output samples, for example, it specifies the destination memory start address at which the output samples have to be written by the output formatter. This is a byte-address but only even values are valid. This register covers the entire address space of the eight local memories (4 × 16KB = 64 KB). The four accelerator local memories are contiguous in the memory address space and any of them can act as the destination memory (as long as the same memory bank is not configured to be used as source memory at the same time).



**Table 24-3. Output Formatter Registers (continued)**

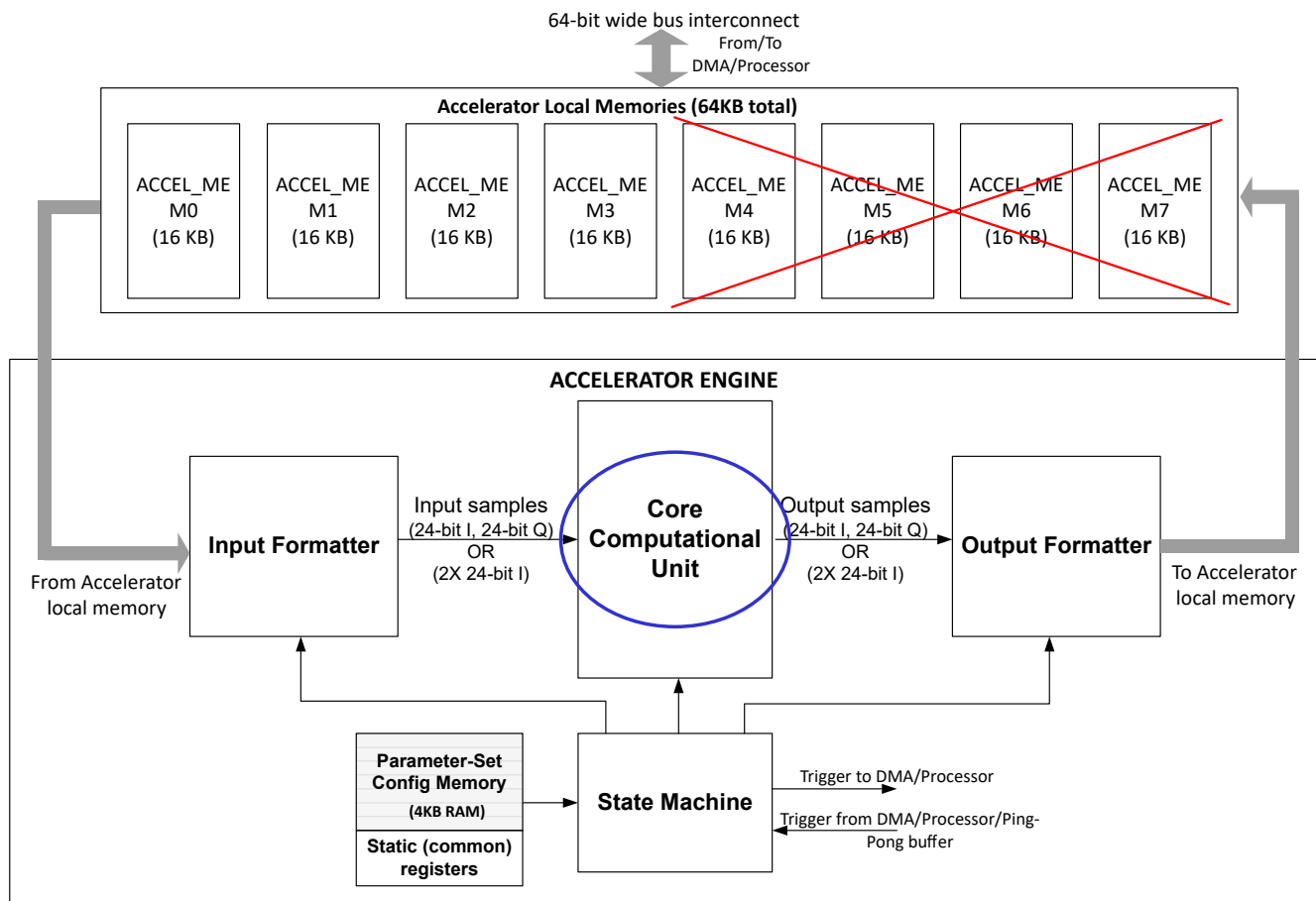
Register	Width	Parameter Set	Description
DSTACNT	12	Yes	Destination sample count: This register specifies the number of samples (minus 1) to be written to the destination memory for every iteration. The sample count is in number of samples, not number of bytes. For example, the sample count can be specified as 191 (DSTACNT = 0x0BF) in a case where 192 samples must be written. Note that the DSTACNT register can be different from SRCACNT or even the FFT size. This is useful when only a part of the FFT bins must be written to memory and the remaining (far-end FFT bins) can be discarded. This register description is true when the DST_SKIP_INIT register value is zero (see further for more information related to DST_SKIP_INIT).
DSTAINDX	16	Yes	Destination sample index increment: This register specifies the number of bytes separating successive samples to be written to the destination memory. For example, a value of DSTAINDX = 16 means that successive samples written to the destination memory should be separated by 16 bytes. Only even values are allowed for this register. The maximum value allowed for this register is 65534.
DSTBINDX	16	Yes	Destination offset per iteration: This register specifies the number of bytes separating the starting address of output samples for successive iterations. For example, when using four iterations to process four RX chains, this register can be used to specify the offset in the starting address between the successive RX chains. Note the distinction that DSTAINDX specifies the number of bytes separating successive samples for a particular iteration, whereas SRCBINDX specifies the number of bytes separating the starting address of the first sample for successive iterations. Only even values are allowed for this register. The maximum value allowed for this register is 65534.
DST_SKIP_INIT	10	Yes	Destination skip sample count: This register specifies how many output samples should be skipped in the beginning, before starting to write to the destination memory. This is useful if only a certain part of the FFT output (skipping the first several bins) need to be stored in memory. The total number of samples written to destination memory is equal to DSTACNT+1-DST_SKIP_INIT.
DSTREAL	1	Yes	Complex or real output: This register-bit specifies whether the output samples are real or complex. A value of DSTREAL = 0 implies complex output and a value of DSTREAL = 1 implies real output. When real output is selected, the output formatter block automatically stores only the real part into the destination memory. This is useful when the core computational unit is configured to output magnitude or log-magnitude values.
DST16b32b	1	Yes	16-bit or 32-bit output word alignment: This register-bit specifies whether the output samples are to be written as 16-bits or 32- bits wide in the destination memory. A value of DST16b32b = 0 implies that the output samples are to be written as 16-bit words (in case of complex output, real and imaginary parts are each 16 bits wide). A value of DST16b32b = 1 implies that the output samples are 32-bits wide each.

**Table 24-3. Output Formatter Registers (continued)**

Register	Width	Parameter Set	Description
DSTSIGNED	1	Yes	Output sign-extension mode: This register-bit, when set, specifies that the output samples are signed numbers and therefore, sign-extension or signed-saturation at the MSB is required when converting the 24-bit wide samples coming from the core computational unit into 16-bit or 32-bit output words to be written to the destination memory.
DSTCONJ	1	Yes	Output conjugation: This register-bit specifies whether the output samples must be conjugated before writing them into the destination memory. If DSTCONJ is set, then the output samples are conjugated. Setting this register-bit only makes sense if the samples are complex numbers (for example, DSTREAL = 0). This register, together with its counterpart in the output formatter block, enables an IFFT mode for the FFT engine.
DSTSCAL	4	Yes	Output scaling: This register specifies a programmable scaling using bit-shift, when converting the 24-bit samples coming from the core computational unit into 16-bit or 32-bit wide words to be written to the destination memory. See Figure 11 and its description for more details regarding this register.
DSTIQSWAP	1	Yes	IQ Swapping : Swap the I & Q samples written out to memory.
IP_OP_FORMATTER_CLIP_STATUS. op_formatter_clip_status	1	No	Read-only register that indicates clip status for output formatter (during scaling).
MEM_INIT_START.MEM_INIT_START_PARAM_RAM	1	No	Writing 1'b1 would start the memory initialization for the Param memory. Self-clearing bit.
MEM_INIT_START.MEM_INIT_START_DMEM3, MEM_INIT_START_DMEM1, .. MEM_INIT_START_DMEM0	1	No	Writing 1'b1 would start the memory initialization for the DMEM3, DMEM2.. DMEM0. Self-clearing bit
DMEM0, DMEM1... DMEM3 (MEM_ACCESS_ERR_STATUS) MEM_ACCESS_ERR_STATUS.dmem0..3	1	No	Memory access error: This set of 4 1-bit read-only registers indicates if there is a memory access error caused by incorrect configuration or usage of the accelerator, where both the DMA and the accelerator are attempting to access the same 16KB memory at the same time. The composite 4-bit register indicates the error status for the 4 16KB memories (DMEM0 bit corresponds to ACCEL_MEM0).

### 24.1.5 Accelerator Engine – Core Computational Unit

This section describes the core computational unit present in the accelerator engine (see Figure 24-12).



**Figure 24-12. Core Computational Unit**

#### 24.1.5.1 Core Computational Unit

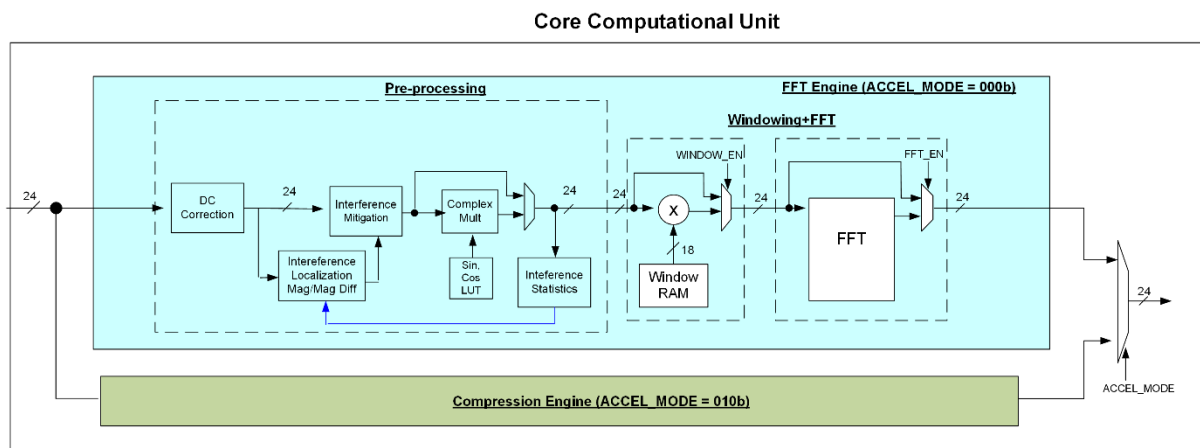
The core computational unit performs the mathematical operations required for the key functions, such as DC & Interference correction, FFT, and compression. The core computational unit accepts a streaming 24-bit complex input (24 bits for each I and Q) from the input formatter block and it outputs a streaming 24-bit complex output (24 bits for each I and Q) to the output formatter block. 24-bit complex input could also be packed with two real inputs improving the throughput.

Figure 24-12 shows the block diagram of the core computational unit. The core computational unit has two main computation engines, namely:

- FFT Engine: Performs Pre-processing, Windowing, FFT.
- Compression Engine: Used for compression and decompression of radar data.

Only one of these two engines can be operational at any given instant. However, in separate parameter sets, different engines can be configured and used, so that multiple parameter sets executing one after another can accomplish a sequence of computational operations as desired.

The register ACCEL\_MODE controls which engine gets used in a given parameter set.



**Figure 24-13. Core Computational Unit Block Diagram of HWA 1.5**

For the purpose of part one of the user's guide, only the FFT Engine path is described. Specifically, the windowing & FFT, covered in this document. The greyed-out blocks in Figure 24-13, namely the Pre-processing is covered in part two of the user's guide and can be ignored for the present purpose. Compression is covered in part three.

#### 24.1.5.1.1 Core Computational Unit – FFT Engine – Operation

The core computational unit operates on the streaming input of samples coming from the input formatter block, and in general outputs a stream of samples (after an initial latency in some cases) to the output formatter block. In general, at steady-state, one complex input sample is processed and one complex output sample is produced every clock. The complex input can be configured to pack two real samples forming a pseudo-complex samples with register 2XMODE = 1.

The FFT Engine in the core computational unit has the ability to perform pre-processing, windowing and FFT. Each of these computational subblocks operate on a streaming input and produce a streaming output at the throughput of one complex or pseudo-complex sample per clock. These computational subblocks are stitched together one after the other in a series, as shown in Figure 24-13. This architecture allows multiple operations to be done in a streaming manner (for example, windowing and FFT can be done together), while at the same time, providing the user flexibility to choose one operation at a time.

The parameter-set registers WINDOW\_EN and FFT\_EN control the multiplexers (see Figure 24-13), which decide what operations are performed on the input samples for that parameter set.

Notethat for the purpose of part one of the user's guide, the register ACCEL\_MODE must be kept at zero. The purpose of this register is covered in part two.

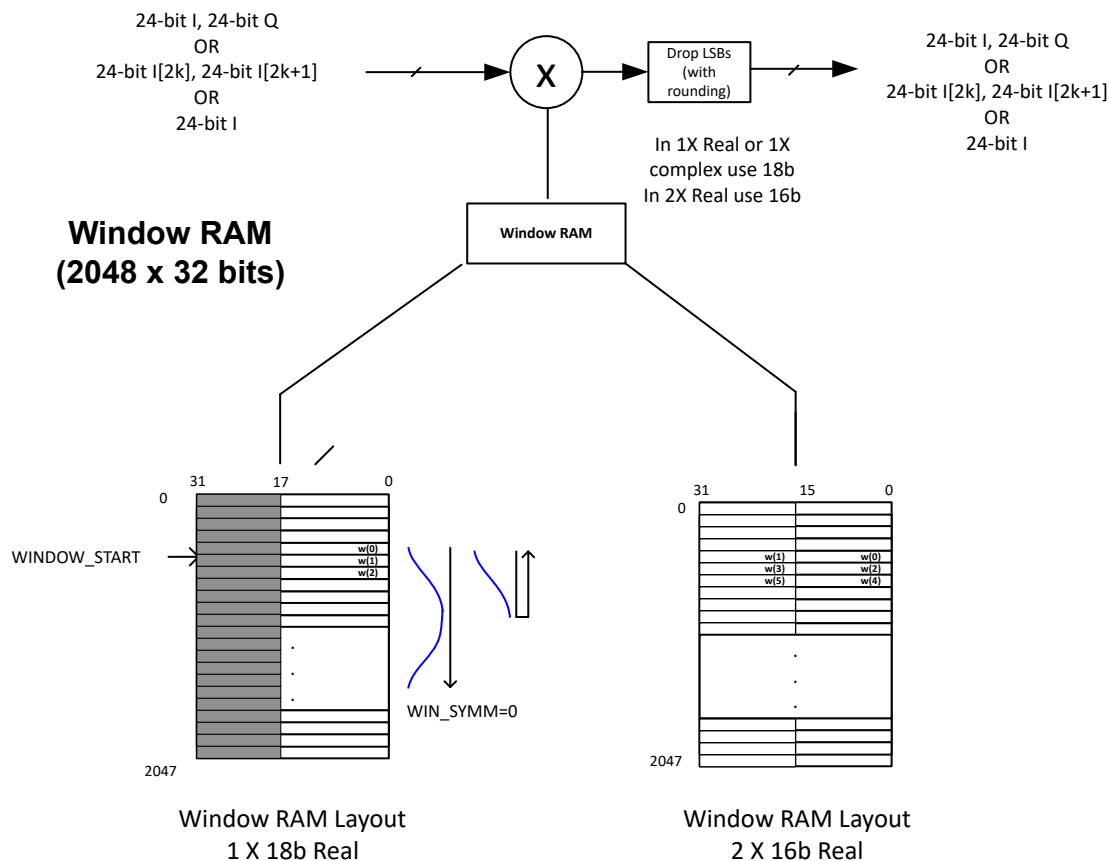
#### 24.1.5.1.2 Core Computational Unit – FFT Engine – Windowing

The incoming samples from the input formatter to the core computational unit are passed through the (optional) windowing operation (see Figure 24-15). Windowing operation is often required prior to performing FFT, to mitigate the sinc roll-off leakage from one strong FFT bin to the adjacent bins.

As the incoming samples from the input formatter stream in, each sample is multiplied by the appropriate window coefficient read from a Window RAM. Because the incoming samples are complex 24-bits wide (24-bits for each I and Q or 24-bits for each  $I_{\text{even}}$  and  $I_{\text{odd}}$ ), the windowing operation involves multiplying the 24-bit I and 24-bit Q of the incoming sample with the window coefficient (see Figure 24-14). The output of this multiplication is rounded back to 24-bit I and 24-bit Q by dropping excess LSBs. Note that windowing can be enabled by setting the register bit WINDOW\_EN to 1.

The window RAM can hold up to 2048 32-bit words. The window coefficients can be stored in these words in one of the following three formats:

1. **18-bit real** coefficients: If WINDOW\_MODE = 0b00, the window coefficients are assumed to be 18-bit signed real values. Up to 2048 real coefficients can be stored in the window RAM in this mode (one 32-bit word in the RAM stores one 18-bit coefficient).
2. **16-bit real** coefficients: If WINDOW\_MODE = 0b01 & 2XMODE = 1, the window coefficients are assumed to be 16-bit signed real values. Up to 4096 real coefficients can be stored in the window RAM in this mode (one 32-bit word in the RAM stores two successive 16-bit coefficients: the 16LSBs store coefficients 0, 2, 4, etc. and 16MSBs store coefficients 1, 3, 5, etc.).



**Figure 24-14. Window RAM Layout for 1X 18b Real and 2X 16b Real modes**

The 18b real coefficients mode can be applied with register SRCREAL = 0 or 1, 2XMODE = 0, while the 16b real coefficients mode is meant to be used with SRCREAL = 0 & 2XMODE = 1.

The start location (32-bit word index) in the window RAM is programmed in a 11-bit register WINDOW\_START as part of the parameter set, so that the windowing computation can pick the appropriate window coefficients starting from that index. For each incoming sample, the index keeps incrementing, so that each successive sample is multiplied by the successive window coefficient. At the end of each iteration (for example, when SRCACNT number of samples have been processed), the index resets back to the starting coefficient index programmed for the parameter set, so that the next iteration can be performed. At the end of all the iterations of the current parameter set, the next parameter set can use a different window if desired. For example, when performing second- and third-dimension FFTs one after another (in two parameter sets), the window functions for both these FFTs can be pre-stored in the Window RAM and appropriate start index can be provided for each of the FFT operation dimensions.

If the window function is symmetric, the user may store only one half of the window coefficients in the Window RAM. The register bit WINSYMM, when set, indicates that after SRCACNT / 2 samples (or, if SRCACNT is odd, (SRCACNT + 1) / 2 samples) are processed, the window coefficients read-indexing must be reversed, so that the same set of coefficients used for the first SRCACNT / 2 samples are reused in the reverse order for the next

SRCACNT / 2 samples. (See [Figure 24-15](#)). If SRCACNT is odd, then the last window coefficient is read twice, when the direction is reversed. If SRCACNT is even, then the last window coefficient is only read once, when the direction is reversed.

In special case of 2XMODE = 1, then irrespective of SRCACNT being odd/even, the last real window coefficient is repeated twice. The output of the windowing computation is 24-bit I and 24-bit Q or 24-bit  $I_{\text{even}}$  and 24-bit  $I_{\text{odd}}$ , which is streamed into the FFT subblock.

#### 24.1.5.1.3 Core Computational Unit – FFT Engine – FFT

The FFT subblock performs complex-input FFT on the incoming 24-bit I and 24-bit Q data stream. The complex FFT size is programmable, in the range, 2 to 2048. FFTs of length  $2^N$  for  $N = 0$  to 11, and  $3 \times 2^N$  for  $N = 0$  to 9 are supported. It also seamlessly performs real input FFT on the incoming 24-bit  $I_{\text{odd}}$  and 24-bit  $I_{\text{even}}$  data stream. The real FFT size is programmable, in the range, 4 to 4096. FFTs of length  $2^N$  for  $N = 2$  to 12, and  $3 \times 2^N$  for  $N = 2$  to 10 are supported

The lowest FFT size of 2 is mostly useful as a *complexadd-subtract* feature. This is however not supported with 2XMODE = 1.

The FFT operation can be enabled or disabled by using the register bit FFT\_EN. When enabled, the FFT subblock computes the FFT of the input data stream and produces a 24-bit I and 24-bit Q output stream. This output stream is initially in bit-reversed order, but the output formatter handles appropriately writing the output to the destination memory in the correct order.

The FFT implementation comprises a series of butterfly stages (for complex input streams) followed by an optional complex conjugate symmetric combiner stage for seamless computation of real-input FFT. Depending on the FFT size needed, an appropriate number of butterfly stages are employed.

The FFT size is programmed using the registers, FFTSIZE and FFTSIZE\_3X\_EN. For power-of-2 FFT sizes, the register bit FFTSIZE\_3X\_EN should be kept 0. The FFT size is configured using the register FFTSIZE and the actual FFT size in this case is  $2^{\text{FFTSIZE}}$ . When an FFT size of the form ( $3 \times \text{power-of-2}$ ) is needed, then the register bit FFTSIZE\_3X\_EN should be set to 1. In this case, the actual FFT size is  $3 \times 2^{\text{FFTSIZE}}$ . The 2XMODE = 1 setting performs an effective real FFT as shown in table below.

**Table 24-4. Config Registers**

SRCACNT	FFTSIZE	FFTSIZE_3X_EN	2XMODE	Effective FFT
31	5	0	0	32-pt complex
95	5	1	0	96-pt complex
15	4	0	1	32-pt real
47	4	1	1	96-pt real

When N-point FFTs of real signals are computed using 2XMODE = 1, the FFT engine produces FFT output only for indices . This is because of the conjugate symmetric property of FFT of real valued signals, and for are redundant - does not contain any valuable information.

In the case of real signals, the FFT samples and are purely real and the user is given one of three options to pack or retain these samples using the 2-bit register CONJCOMB\_OUTMODE. When set to 0, output memory value for sample index zero is set to . When set to 1, output memory value for sample index zero is set to . When set to 2, output memory value for sample index zero is set to .

Note that the FFT size must be equal to or larger than SRCACNT, and the input formatter block automatically zero-pads extra samples to account for the difference between FFT size and SRCACNT.

#### 24.1.5.1.4 Core Computational Unit – FFT Engine – FFT – Zero Padding

The FFT engine has provision for *zeropadding*, which is important when performing FFT of a set of samples whose length doesn't match a supported FFT size. The FFT engine automatically feeds the required number

of zeros into the core computational unit, whenever the FFT size (as programmed using the FFTSIZE register, which is described in a later section) does not match the SRCACNT setting.

For example, if the number of input samples read by the input formatter is 56 (for example, SRCACNT = 55) and the FFT length is to be 64 (FFTSIZE = 6, FFTSIZE\_3X\_EN = 0), then the FFT engine feeds 8 zeros at the end of each iteration, before starting to read the input samples for the next iteration from the source memory. This zero-padding provision enables the core computational unit to perform 64-point FFT with the correct set of zero-padded input samples. With same settings and 2XMODE = 1, the FFT engine effectively performs a 128-pt FFT and appends zeros (8 zero-input pairs)

The zero padding is effective only when performing FFT operation in the core computational unit (i.e., when FFT\_EN = 1) and not otherwise.

#### 24.1.5.1.5 Core Computational Unit – FFT Quantization and Speed Performance

As is well known, a butterfly stage typically consists of add-subtract and twiddle multiplication operations. At the output of each add-subtract structure, the bit-width would increase by 1 bit (for example, 24-bit input would grow to 25-bit output). To handle this one-bit growth due to add-subtract operation, there is a provision at the output of each butterfly add-subtract stage to scale the result back to 24 bits, by either dividing the output by 2 (round off one LSB) or by saturating one MSB, shown in [Figure 24-15](#).

The multi-bit register BFLY\_SCALING is used to control this divide-by-2 scaling operation at each stage, so that the user has full flexibility to control the signal level through the different butterfly stages. If BFLY\_SCALING = 0 for a particular stage, then the 25-bit output is saturated at the MSB to get back to 24 bits. Otherwise, it is convergent-rounded at the LSB to get back to 24 bits. The user can thus control the scaling at each of the butterfly stages. The LSB of this multi-bit register corresponds to the last stage and the MSB of this register corresponds to the first stage. For an FFT size of 64, only the LSB 6 bits are relevant. Similarly, for an FFT size of 3\*64, the LSB 6 bits are relevant. Additionally, the register, BFLY\_SCALING\_FFT3X indicates the scaling option for the single radix-3 stage (i.e., it supports removing 0 or 1 or 2 LSBs). The Conjugate Symmetric Combiner stage consists of 2 back-to-back add-subtract operation with a twiddle multiplication operation, and the output bit-width would increase by 2 bits (24-bit input to 26-bit output). For an FFT size of 64 with 2XMODE enabled (used to compute 128-point real FFT), the LSB 6 bits of BFLY\_SCALING is relevant, along with the register CONJCOMB\_SCALING, that indicates the scaling options for the Conjugate Symmetric Combiner stage (i.e., it supports removing 0 or 1 or 2 LSBs).

There is a multi-bit read-only register FFTCLIP which indicates whether there was any clipping in any of the butterfly stages. This register is a sticky register that gets set when a clipping event occurs and remains set until it is cleared using the CLR\_FFTCLIP register bit. See the register description of FFTCLIP in [Table 24-5](#).



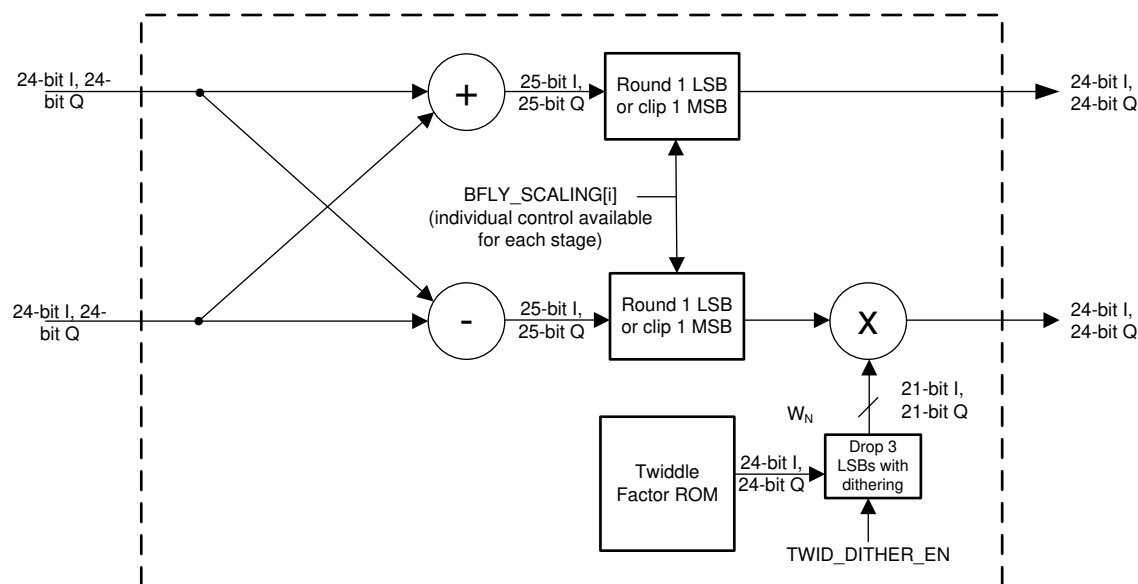


Figure 24-15. Butterfly Stage Fixed-Point

The twiddle factors are stored as 24-bit I and 24-bit Q coefficients. Prior to twiddle factor multiplication, the coefficients are reduced to 21-bit I and 21-bit Q by dropping three LSBs (with optional dithering). The purpose of dithering is to eliminate any repetitive quantization noise patterns from degrading the SFDR of the FFT. The use of dithering here is optional. For dithering, an LFSR is used to generate a random pattern, for which the LFSR seed must be loaded with a non-zero value (see LFSRSEED in the register descriptions).

The SFDR performance of the FFT, with dithering enabled, is better than -140 dBc, as shown in Figure 24-16.

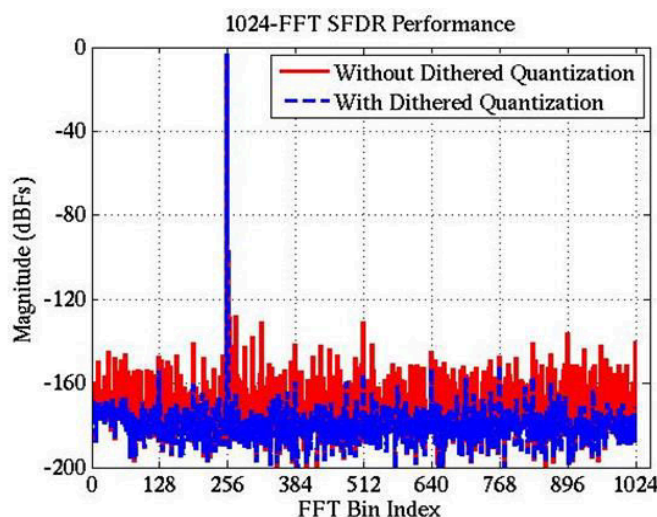


Figure 24-16. FFT SFDR Performance With and Without Dithering

The architecture of the FFT is such that it can take a streaming complex input (one sample per clock) and produce a streaming FFT output (one sample per clock), in steady-state. There is an initial latency of approximately  $FFT\ size$  number of clocks for complex inputs and approximately  $1.25(FFT\ size / 2)$  number of clocks when 2XMODE is enabled for FFT computation of real signals. This latency only comes into picture once for a given parameter set. Within a parameter set, multiple FFT iterations can be performed back-to-back (for example, for four RX) with no additional latency between iterations.



Because the implementation uses 300 MHz in AWR2544 device, a 256-point complex FFT for four RX chains would take  $256 + 256 \times 4$  clock cycles to complete, which corresponds to  $4.27 \mu\text{s}$  (plus a few clocks of implementation latencies, which are not accounted here since it is negligible). In 2XMODE a 256-point real FFT for four RX chains would take , which corresponds to  $2.24 \mu\text{s}$ , is almost twice as fast as that of a complex FFT. [Table 24-5](#) lists the approximate computation time needed for various FFT sizes. It can be noted that when 2XMODE is set to 1, FFTs of real valued signals can be computed at approximately twice the speeds compared to when 2XMODE = 0.

**Table 24-5. Complex and Real FFT Computation Time (for 300MHz clock)**

Example	FFT Size	2XMODE	Numberof Back-to-Back Iterations	Numberof Clock Cycles (Initial latency + Computation)	Total Duration
1	256	0	4	$256 + (256 \times 4)$	$4.27 \mu\text{s}$
2	128	0	4	$128 + (128 \times 4)$	$2.133 \mu\text{s}$
3	8	0	64	$8 + (64 \times 8)$	$1.73 \mu\text{s}$
4	256	1	4	$(128 \times 1.25) + (128 \times 4)$	$2.24 \mu\text{s}$
5	128	1	4	$(64 \times 1.25) + (64 \times 4)$	$1.12 \mu\text{s}$
6	8	1	64	$(4 \times 1.25) + (4 \times 64)$	$0.87 \mu\text{s}$

The output of the FFT is fed to the output formatter.

#### 24.1.5.1.6 Core Computational Unit – FFT Engine – Register Descriptions

[Table 24-6](#) lists all the registers of the FFT Engine within the core computational unit.

**Table 24-6. FFT Engine Registers**

Register	Width	Parameter Set	Description
2XMODE	1	Yes	2X Real input mode: This register-bit conveys whether the input stream is complex or 2X real. The pre-processing chain then interprets the Q-channel of I,Q data-path as additional I channel (even). This bit is only applicable with SRCREAL = 0.
WINDOW_EN	1	Yes	Windowing Enable: This register-bit enables or disables the pre-FFT windowing operation. If this register is set to 1, then the windowing is enabled, otherwise, it is disabled. The exact window function (coefficients) to be applied is specified in a dedicated Window RAM, which is $2048 \times 32$ bits in size.
FFT_EN	1	Yes	FFT Enable: This register-bit is used to enable the FFT computation. If FFT_EN = 1, then the FFT computation is enabled. Otherwise, it is disabled (bypassed).
WINDOW_START	11	Yes	Windowing coefficients start location in Window RAM: This register specifies the starting location (32-bit word index) of the window coefficients within the Window RAM. The purpose of this register is to allow multiple windows (for example, one window of 512 coefficients and another window of 256 coefficients) to be stored in the Window RAM and one of these windows can be used by programming this start location register appropriately in the current parameter set.
WINSYMM	1	Yes	Window symmetry: This register-bit indicates whether the complete set of window coefficients are stored in the Window RAM or whether one half of the coefficients are stored. If this register bit is set, it means that the window function is symmetric and therefore, only one half of the window function coefficients are stored in the Window RAM. See the description section related to Windowing computation for more details.
WINDOW_MODE	1	Yes	Window Mode: 0b : 18-bit signed real coefficients, 1b: 16-bit signed real coefficients
FFTSIZE	4	Yes	FFT size: This register indicates the number of FFT radix-2 butterfly stages employed. Refer detailed description section for more details on this register.

**Table 24-6. FFT Engine Registers (continued)**

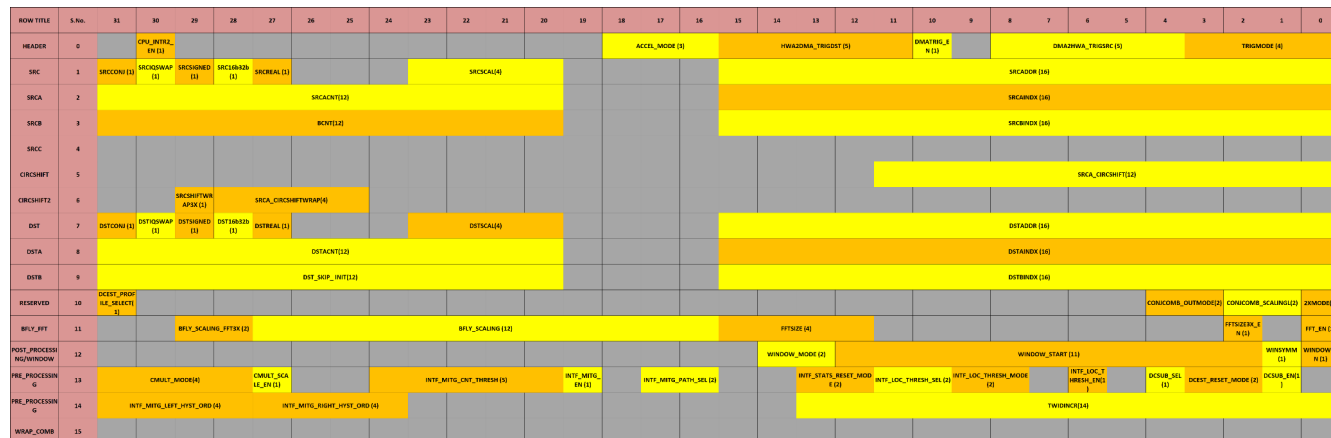
Register	Width	Parameter Set	Description
FFTSIZE3X_EN	1	Yes	FFT size 3X enable: This register indicates whether to engage the additional FFT radix-3 butterfly stage. Together with FFTSIZE, this register specifies the FFT size. This can be used to realize FFTs of length $3 \times 2^N$ point FFTs..
BFLY_SCALING	12	Yes	Butterfly scaling for radix-2 stages: This register is used to control the butterfly scaling at each of the radix-2 butterfly stages. If the Nth bit in this register is set to 0, then the 25-bit output of the Nth radix-2 stage from the last is saturated to 24-bit. Otherwise it is scaled down by 2 and rounded to produce a 24-bit output.
BFLY_SCALING_FFT3X	2	Yes	Butterfly scaling for radix-3 stage: This register is applicable only if FFTSIZE3X_EN is set to 1. This register is used to control the butterfly scaling in the 3-point FFT structure that precedes the powers-of-2 FFT structure. If this register is set to 0, then that 26-bit output after radix-3 stage is saturated at the MSB to get back to 24 bits. If it is 2, it is rounded to remove 2 LSBs to get back to 24 bits. A middle option exists by setting this to 1. In this case, the 26 bit temporary output is convergent-rounded to remove 1 LSB and the 25-bit output thus obtained is saturated to 24 bits.
CONJCOMB_SCALING	2	Yes	Scaling option for Conjugate Symmetric Combiner Stage: This register is applicable only if 2XMODE is set to 1. This register is used to scale the conjugate symmetric combiner stage that succeeds the powers-of-2 FFT structure. If this register is set to 0, then that 26-bit output is saturated at the MSB to get back to 24 bits. If it is 2, it is rounded to remove 2 LSBs to get back to 24 bits. A middle option exists by setting this to 1. In this case, the 26-bit temporary output is convergent-rounded to remove 1 LSB and the 25-bit output thus obtained is saturated to 24 bits.
CONJCOMB_OUTMODE	2	Yes	Register to handle the real FFT or Conjugate Symmetric Combiner output and . This register is applicable only if 2XMODE is set to 1. When set to 0, output memory value for sample index zero is set to . When set to 1, output memory value for sample index zero is set to . When set to 2, output memory value for sample index zero is set to . (Refer section 5.1.3)
DITHER_TWID_EN. dither_twid_en	1	No	Twiddle factor dithering enable: This register-bit is used to enable and disable dithering of twiddle factors in the FFT. The twiddle factors are 24-bits wide (24-bits for each I and Q), but they are quantized to 21-bits before twiddle factor multiplication. This quantization is implemented with dithering on the LSB, to avoid periodic quantization pattern affecting SFDR performance of the FFT. It is recommended to keep this register bit set to 1 (dithering enabled), with appropriate LSFR seed loaded.
LFSR_SEED. lfsr_seed	29	No	Seed for LFSR (random pattern): For twiddle factor dithering, there is an LFSR that is used, whose seed value is loaded by writing to this 29-bit LFSRSEED register. The LFSRSEED register should be set to any non-zero value, say 0x1234567. To load the LFSR seed, a pulse signal needs to be provided, by writing a 1 followed by a 0 (i.e., by setting and clearing) the LFSRLOAD register-bit.
LFSR_LOAD. lfsr_load	1	No	For twiddle factor dithering, there is an LFSR that is used, whose seed value is loaded by writing to this 29-bit LFSRSEED register. The LFSRSEED register should be set to any non-zero value, say 0x1234567. To load the LFSR seed, a pulse signal needs to be provided, by writing a 1 followed by a 0 (i.e., by setting and clearing) the LFSRLOAD register-bit.

### Table 24-6. FFT Engine Registers (continued)

Register	Width	Parameter Set	Description
FFT_CLIP.fft_clip	14	No	FFT Clip Status (read-only): This is a read-only status register, which indicates any saturation/clipping events that have happened in the FFT butterfly stages. Note that each of the individual butterfly stages in the FFT can be programmed to either saturate the MSB or round the LSB. Whenever saturation of MSB is used in any stage, there is a possibility that that stage can saturate or clip samples. In that case, this saturation event is indicated in the corresponding bit in this status register, so that the processor can read it. If multiple FFTs are performed, this status register includes any saturation events happening in any of them. This status register can only be cleared by the R5F, by setting another single-bit register CLR_FFTCLIP, so that the saturation status indication gets cleared back to 0 and any subsequent saturation events can be freshly monitored. The MSB of this register indicates clip status corresponding to the Conjugate Symmetric Combiner stage. The (MSB-1) of this register indicates clip status corresponding to the radix 3 butterfly (note: it is the MSB and (MSB-1), independent of the number of radix-2 stages engaged).
CLR_FFTCLIP.clr_fftclip	1	No	Clear FFT Clip Status register: This register bit, when set, clears the FFTCLIP register.
WINDOW_RAM[2048] DSS_HWA_WINDOW_RAM	32b each	No	This RAM stores the window co-efficients. Note that there is only one RAM and based on WINDOW_MODE, the samples are accordingly chosen as illustrated in Fig. 15
MEM_INIT_START. MEM_INIT_START_WINDOW_RAM	1	N	Writing 1'b1 would start the memory initialization for the window memory. This bit is self-clearing.
ACCEL_MODE	3	Yes	Select Core Computational Unit Data Path: This register selects the data-path of the accelerator's core computational unit – for example, it selects which of the paths: the FFT engine path, or the compression path, or none is active. Value = 0b000: FFT engine path Value = 0b010: Compression / decompression engine path Value = 0b111: No Operation. The No Operation setting can be used together with an appropriate trigger mode to cause the state machine to wait for an event before moving to the next parameter-set.

### 24.1.6 Appendix A

The parameter-set register layout is provided below for all the accelerator modes. Note that the parameter-set RAM must be written using 32-bit word writes only (i.e., byte-writes and half-word writes are NOT supported).

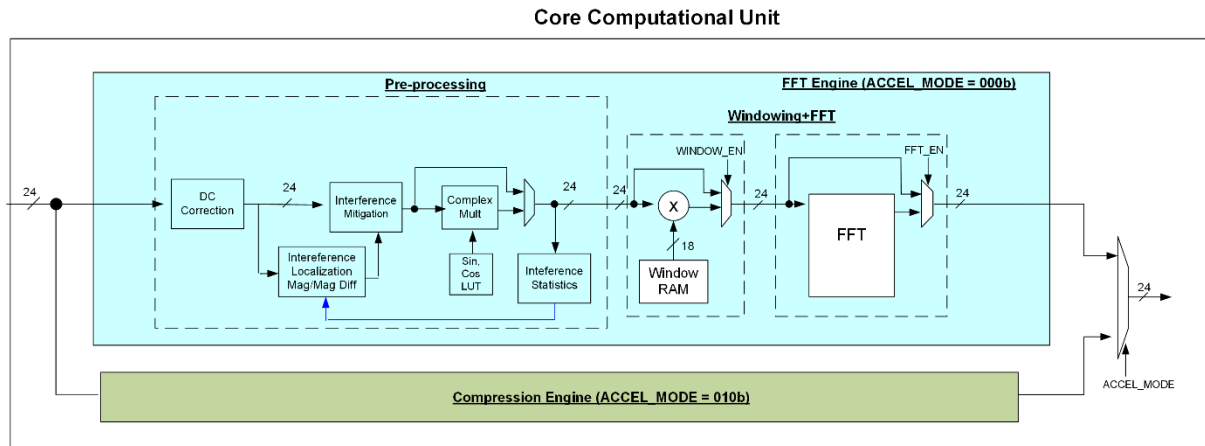


**Figure 24-17. FFT Path Parameter Set Layout**



- The first part of the user guide provides an overview of the overall architecture and features available in the radar hardware accelerator. The main features, such as, input/output formatter, windowing and FFT are covered in the first part.
- The second part of the user guide covers additional features like DC Estimation and correction, Interference localization/mitigation and input pre-scaling. This part of the user's guide assumes that the user has already read and understood Part 1 (see Radar Hardware Accelerator 1.5 User's Guide – Part 1).
- The third part of the user guide covers compression engine.

As explained in Part 1 of the user guide, the FFT Engine comprises pre-processing, windowing and FFT computation subblocks and these are stitched together one after the other in series (see [Figure 24-19](#)). This architecture allows multiple operations to be done in a streaming manner (for example, windowing and FFT can be done together), while at the same time, providing the user flexibility to choose one operation at a time. This section provides an overview of the pre-processing subblock inside the FFT engine of the core computational unit



**Figure 24-19. Core Computational Unit**

The pre-processing subblock provides capability for DC estimation and correction, Interference localization and mitigation, complex multiplication and interference statistics.

#### 24.2.1.1 DC Estimation

The DC estimation block estimates the time-domain average of the stream of samples along the A dimension. The stream can be one chirp, or set of chirps, i.e., frame. The DC is estimated on a per-iteration basis (i.e., along A dimension for each B iteration) for I & Q samples. This is supported only for 8 independent channels in HWA 1.5 unlike 12 in HWA 2.1. These 8 independent channels (0..7) map to BCNT 0..3 in two different profiles – 0 & 1.

DC estimation is based on accumulation followed by a fine scaling and a programmable right shift. The fine scaling is configured as a “1.8” fixed-point value, via the 9-bit scale register. The subsequent programmable right shift is configurable from 6 to 20 bits. Therefore, the DC estimation is well suited for cases where the number of samples per iteration is between  $2^6$  and  $2^{20}$ .

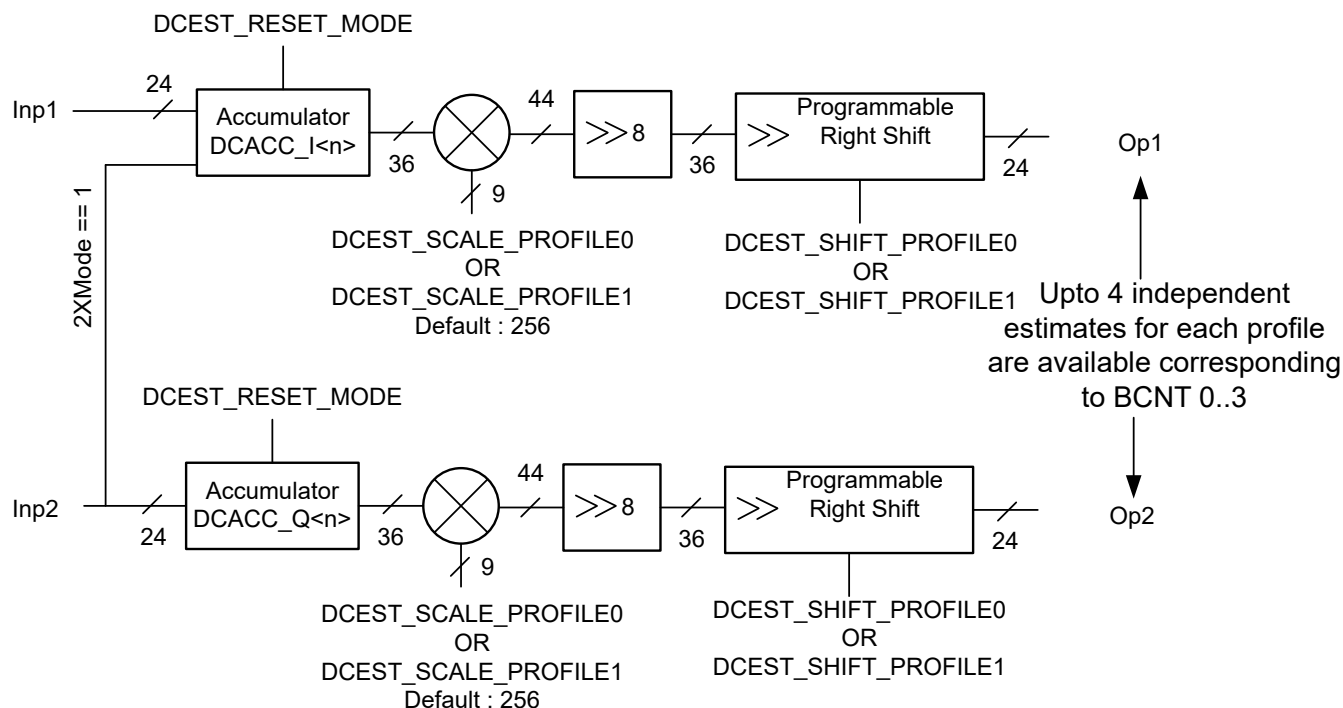
Two sets of shift & scale values are possible based on DCEST\_PROFILE\_SELECT bit–

- Value 0'b (Profile 0) : DCEST\_SHIFT\_PROFILE0, DCEST\_SCALE\_PROFILE0 which provide DCESTI\_0VAL, DCESTI\_1VAL, DCESTI\_2VAL, DCESTI\_3VAL & DCESTQ\_0VAL, DCESTQ\_1VAL, DCESTQ\_2VAL, DCESTQ\_3VAL.
- Value 1'b (Profile 1): DCEST\_SHIFT\_PROFILE1, DCEST\_SCALE\_PROFILE1 which provide DCESTI\_4VAL, DCESTI\_5VAL, DCESTI\_6VAL, DCESTI\_7VAL & DCESTQ\_4VAL, DCESTQ\_5VAL, DCESTQ\_6VAL, DCESTQ\_7VAL. Note that BCNT is still 0..3.

DC estimation works for both real and complex inputs. In particular for real-only inputs, with 2XMODE = 1, DC estimation operates with 2X input throughput. The fixed-point details are captured in [Figure 24-20](#). The internal accumulator reset supports several modes as shown in [Table 24-7](#). For example, when DCEST\_RESET\_MODE = 2, the internal DC accumulators are reset at the beginning of the current parameter-set execution. Therefore, this mode estimates DC value for each set of SRCACNT samples along the A-dimension for up to 4 iterations (per profile) along B-dimension within the current parameter-set. This mode is useful for per-chirp DC estimation. In this mode, the estimated DC values per iteration are latched at the end of current param-set and the accumulators are reset at the start. On the other hand, when DCEST\_RESET\_MODE = 3, the internal DC accumulators are reset only when the state machine executes the first loop of the parameter-set. As the state machine loops through various parameter-sets multiple times as programmed via NLOOPS register, the DC accumulators are not reset in between these loops. This mode is useful for per-frame DC estimation, where each loop corresponds to one chirp and the NLOOPS loops (chirps) correspond to a complete frame. The estimated DC values per iteration are latched at the end of last execution of the param-set.

The processor can read the DC estimates through the read-only registers – DCESTI\_0VAL, ..., DCESTI\_7VAL & DCESTQ\_0VAL, ..., DCESTQ\_7VAL. The DC estimates can also be used for DC subtraction described next.

Mode	Inp1	Inp2	Op1	Op2
2X Real	I[2k]	I[2k+1]	DCEST_I	0
1X Real	I[k]	0	DCEST_I	0
1X Complex	I[k]	Q[k]	DCEST_I	DCEST_Q



**Figure 24-20. DC Estimation**

**Table 24-7. DC Estimation – Reset Modes**

DCEST_RESET_MODE	Comments
0	Hold the DC internal accumulators without updating (bypass DC estimation).
1	DC estimation enabled, but free-running without automatic reset (i.e., not reset at the start of this parameter-set). In this mode, the software can reset the DC accumulators by writing to DC_EST_RESET_SW register bit.
2	Reset the DC internal accumulators at the start of this parameter-set. This mode is applicable for per-chirp DC estimation.
3	Reset the DC internal accumulators at the start of this parameter-set only if the loop-counter is 0. This mode is applicable for per-frame DC estimation.

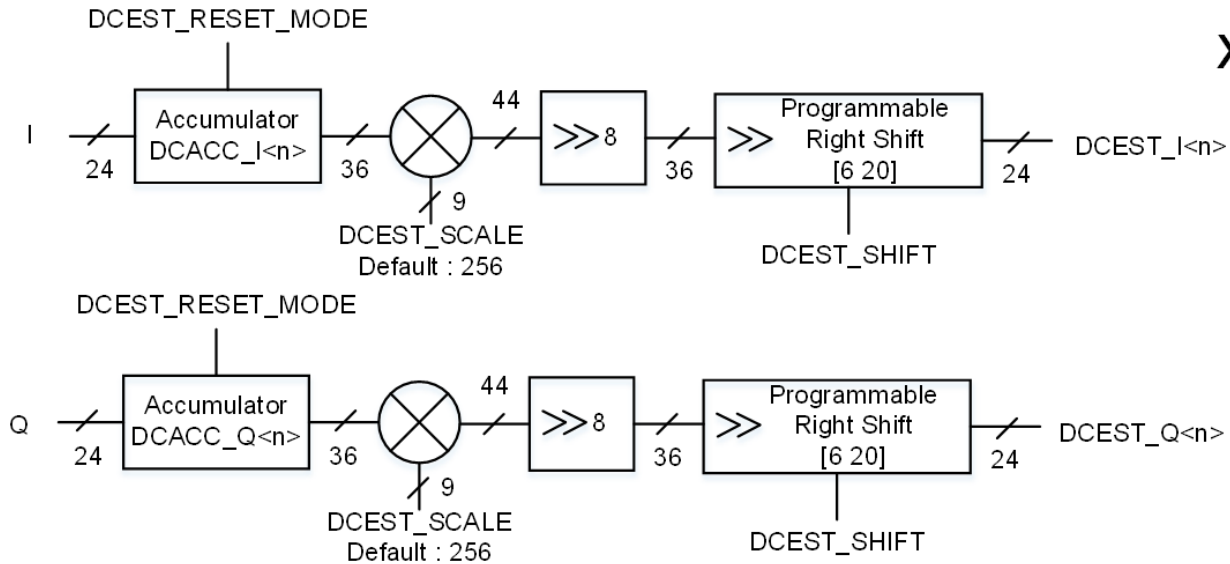


Figure 24-21.

#### 24.2.1.2 DC Subtraction

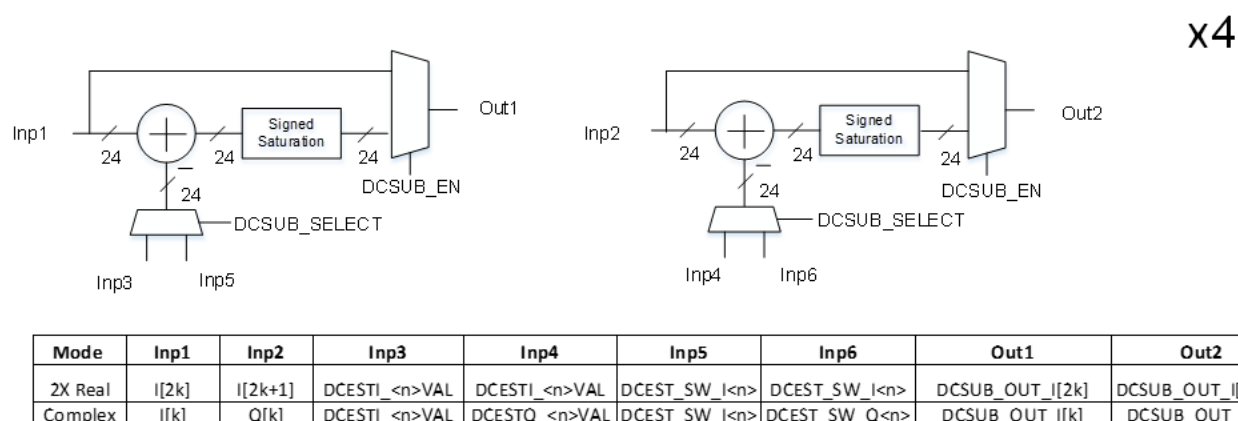
The DC subtraction feature is enabled if the register DCSUB\_EN is set to 1.

DC subtraction (see Figure 24-22) can use the output from the built-in DC estimation accumulators, or a user-programmed value, based on the register bit, DCSUB\_SELECT. If DCSUB\_SELECT is 1, the DC estimation based on the internal accumulators is used. If DCSUB\_SELECT is 0, the software override values are used (they are given by DC\_SW\_I\_<n> and DC\_SW\_Q\_<n> for the n<sup>th</sup> iteration).

For real-only inputs with 2XMODE = 1, the inputs are as indicated in figure below. Note that in the 2XMODE = 1, when DCEST\_PROFILE\_SELECT is 0 & DCSUB\_SELECT is 1, then the internal estimates DCESTI\_0VAL, DCESTI\_1VAL, ..DCESTI\_3VAL are used and when DCEST\_PROFILE\_SELECT is 1, the estimates DCESTI\_4VAL, DCESTI\_5VAL, ..DCESTI\_7VAL are used corresponding to BCNT 0..3.

When using the built-in DC estimation accumulators, DC subtraction is performed on 4 individual streams corresponding to 4 RX on a per-iteration basis (two different sets of 4 values based on profile). Note that in a typical usage, for performing per-chirp DC estimation and DC subtraction, a two-pass approach is needed, where the first pass is configured for DC estimation via one parameter-set, and the second pass is configured for DC subtraction in the next parameter-set. Alternately, if a previous DC estimate (e.g. From the previous chirp) is desired to be used for DC subtraction for the current chirp, then DC subtraction can be directly accomplished in one pass.





**Figure 24-22. DC Subtraction**

### 24.2.1.3 Interference Localization

In an FMCW radar transceiver, interference from another radar typically manifests itself as a time-domain spike in a few samples. This spike corresponds to the time duration when the chirping frequency of both radars overlap with each other. Such a time-domain spike caused by interference can lead to degradation in the noise floor at the FFT output, causing degradation in detection performance.

In order to mitigate the impact of interference, the pre-processing block provides capability to perform interference localization to identify samples corrupted by interference, followed by interference mitigation to repair those samples.

The INTF\_LOC\_THRESH\_EN register is provided as part of the parameter-set to control when the interference localization should be enabled. When enabled, the complex input samples are fed through a magnitude calculation (based on JPL approximation), which computes a 24-bit magnitude of the 24-bit input complex sample. Similarly, magnitude of the backward difference between adjacent samples is also computed, which is another useful metric for interference (glitch) detection. For real only inputs with 2XMODE = 1, only a simple absolute value computation is done.

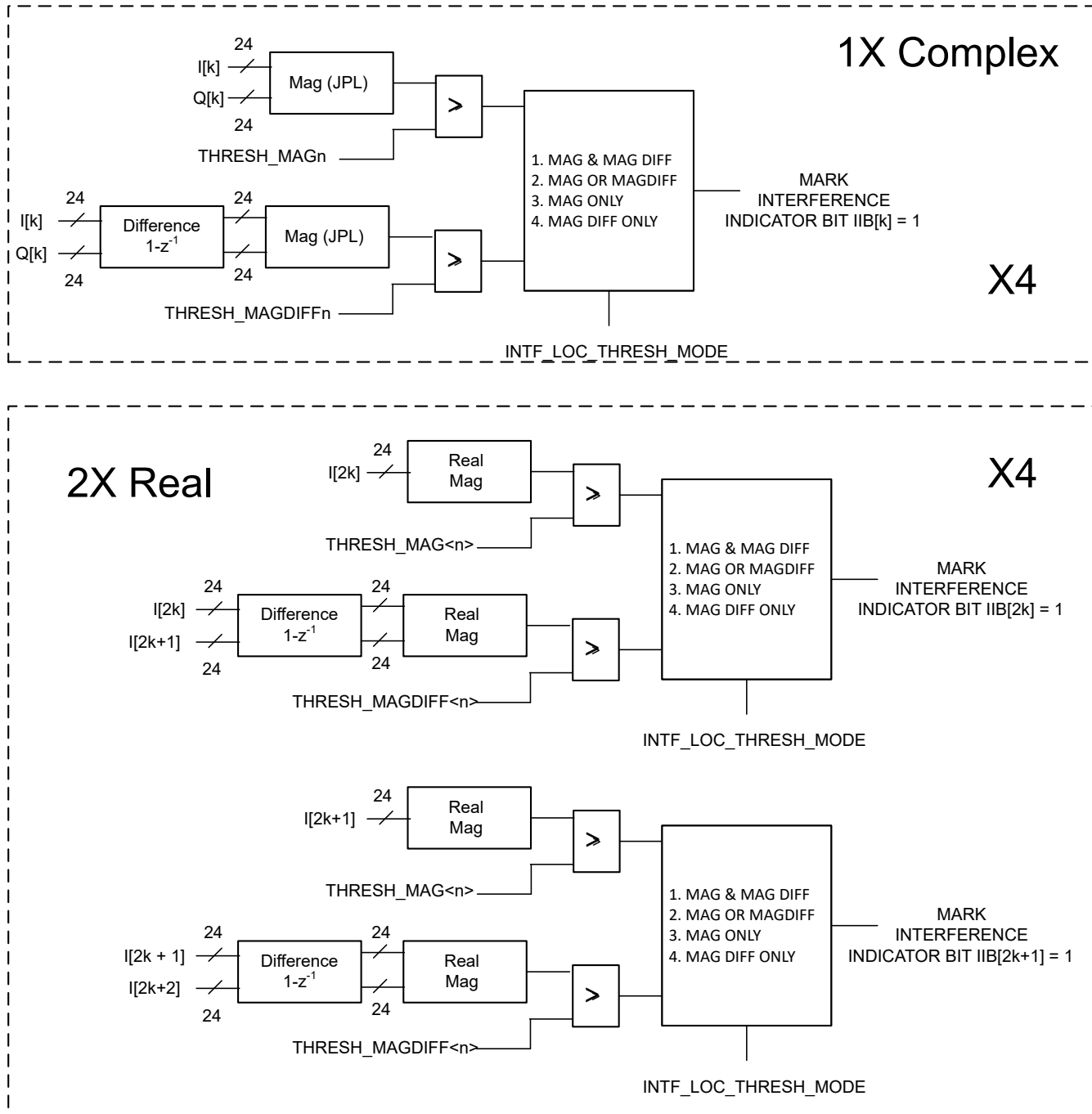
As shown in [Figure 24-23](#), any sample whose magnitude and/or magnitude of backward difference exceeds thresholds THRESH\_MAG<n> and THRESH\_MAGDIFF<n> is considered as affected by interference and is marked by a corresponding Interference indicator Bit (IIB). This is supported individually for up to 4 iterations. Note that the IIB bit is just an internal signal that is set by the interference localization module, in order to mark samples for the interference mitigation module (described later). For real-only inputs, two IIB bits are computed per clock cycle.

The register, INTF\_LOC\_THRESH\_MODE (applies to all iterations) determines the logic to set the IIB bit using the magnitude and/or magnitude of difference estimates. Based on this register, samples are marked with IIB if they exceed the THRESH\_MAG<n>, or THRESH\_MAGDIFF<n>, a logical AND of both, a logical OR of both.

The threshold values of THRESH\_MAG<n> and THRESH\_MAGDIFF<n> applied on a per-channel basis can be derived from SW – INTF\_LOC\_THRESH\_MAG<n>\_SW, INTF\_LOC\_THRESH\_MAGDIFF<n>\_SW, or from a built-in Interference statistics block – INTF\_LOC\_THRESH\_MAG<n>, INTF\_LOC\_THRESH\_MAGDIFF<n> as described in the next section. The user can also choose to sum the built-in interference statistics estimates across all channels to derive a common interference threshold across all iterations– INTF\_STATS\_SUM\_MAG, INTF\_STATS\_SUM\_MAGDIFF. The register, INTF\_LOC\_THRESH\_SELECT is used to select these threshold options. Note that there is unlike DC Estimation, there is no profile selection applicable and <n> maps directly to BCNT 0...3.



The number of samples marked with IIB across the iterations is recorded in the read-only registers, INTF\_LOC\_COUNT\_ALL\_CHIRP and INTF\_LOC\_COUNT\_ALL\_FRAME. This can be read after every chirp or after the completion of a frame (when the state machine completes all the programmed parameter-set loops and enters idle state).



**Figure 24-23. Interference Localization**

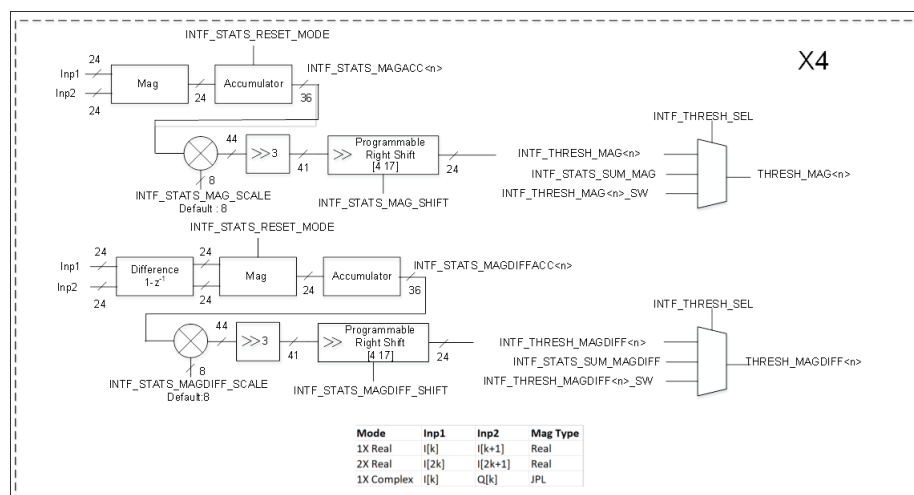
#### 24.2.1.4 Interference Statistics

This block ([Figure 24-24](#)) provides the thresholds for interference localization. In order to obtain the interference statistics and derive the thresholds, the magnitude and magnitude of backward difference of the incoming

samples are accumulated per iteration and up to 4 such independent accumulations are supported. These registers can be reset on a per-chirp or per-frame basis, and the behavior can be controlled using the register INTF\_STATS\_RESET\_MODE. These reset modes are similar to DCEST\_RESET\_MODE previously explained (refer Table 24-7). The interference statistics accumulators can be reset by software via writing the INTF\_STATS\_RESET\_SW register bit. This reset also clears the INTF\_LOC\_COUNTS.

The determination of interference threshold for interference localization is based on taking the above accumulator values and applying a programmable fine scaling, followed by a programmable right shift. The fine scaling is configured via 8-bit registers, INTF\_STATS\_MAG\_SCALE and INTF\_STATS\_MAGDIFF\_SCALE in 5.3 format. The fine scaling value is interpreted as an unsigned 8-bit number with 5 integer bits and 3 fractional bits giving a scale in range [0 to 31.875]. The default value of this register is 8, applying a scaling of 1.0. The programmable right-shift in the range of 6 to 12 is applied via the registers, INTF\_STATS\_MAG\_SHIFT and INTF\_STATS\_MAGDIFF\_SHIFT respectively. Note that if the sum mode of threshold selection is made, then the shift values have to include the extra division based on number of iterations being summed.

The resulting values INTF\_LOC\_THRESH\_MAG<sub>n</sub> and INTF\_LOC\_THRESH\_MAGDIFF<sub>n</sub> are used as thresholds in the interference localization block as described in the previous section. Note that there is unlike DC Estimation, there is no profile selection applicable and <n> maps directly to BCNT 0...3.



**Figure 24-24. Interference Statistics**

### 24.2.1.5 Interference Mitigation

Interference mitigation block (Figure 24-25) uses the results of Interference Localization block and mitigates the interference affecting the input samples which are marked as interference-corrupted through the Interference Indicator Bit. Interference mitigation is applicable for max. 4 iterations

The interference mitigation feature can be enabled by setting the INTF\_MITG\_EN bit in the parameter-set.

The first sub-module of the Interference Mitigation Block is a hysteresis module, which provides de-bouncing logic. For each incoming sample, its own IIB bit, as well as INTF\_MITG\_RIGHT\_HYST\_ORD number of right IIB bits and INTF\_MITG\_LEFT\_HYST\_ORD number of left IIB bits are considered in order to decide whether that particular sample is actually affected by interference. If 2XMODE = 1, then for real inputs only, the INTF\_MITG\_LEFT\_HYST\_ORD and INTF\_MITG\_RIGHT\_HYST\_ORD number of bits are restricted to be even numbers. If the number of IIB bits in that interval is greater than or equal to INTF\_MITG\_CNTTHRESH, then that sample is assumed to be affected by interference. Thus, the hysteresis module outputs a filtered version of the IIB bit stream, which is then used for interference mitigation.

There are three different options (Figure 24-26) for interference mitigation and one of these can be selected using the INTF\_MITG\_PATH\_SEL register. If INTF\_MITG\_PATH\_SEL = 0, then the interference mitigation block

simply zeros out samples that are marked with the IIB bit. This is a simple form of interference mitigation. With 2XMODE = 1, it corrects for two samples per clock

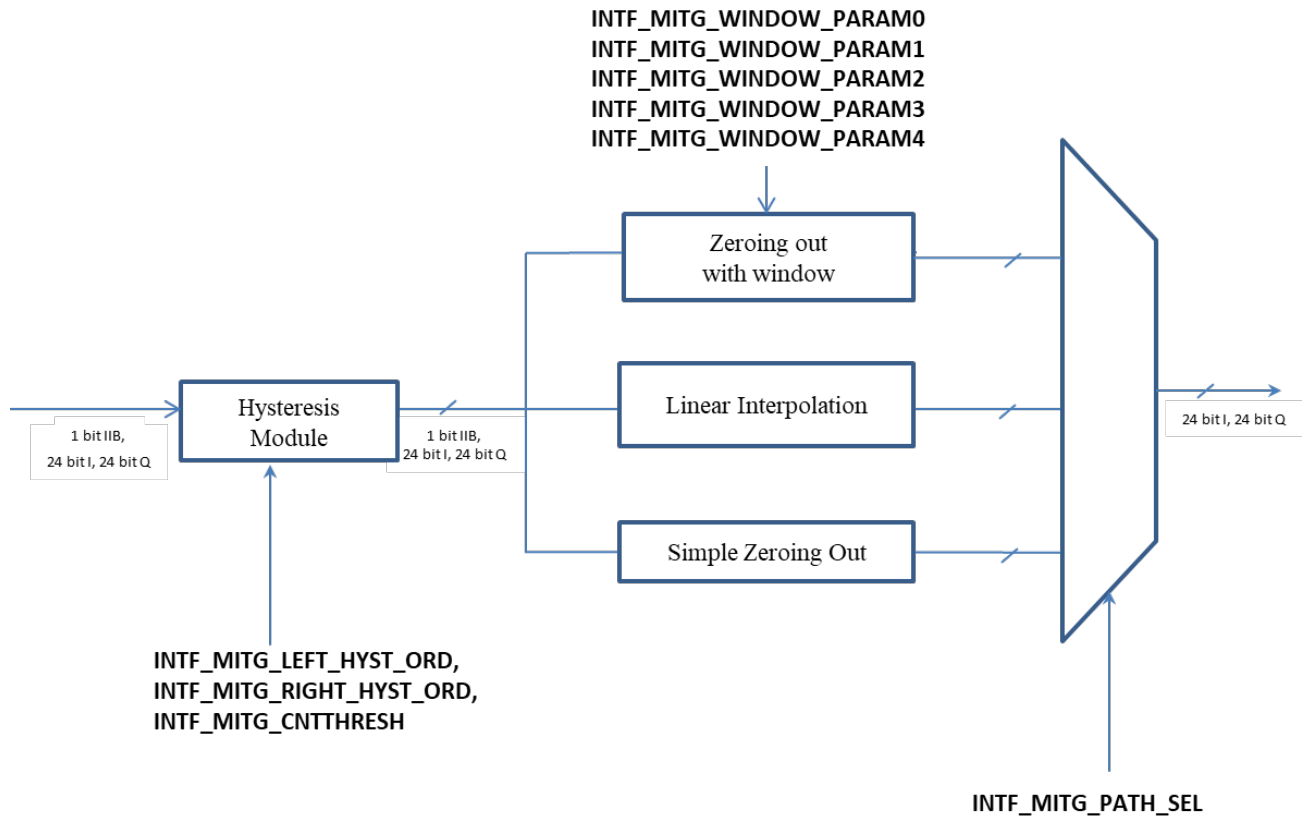
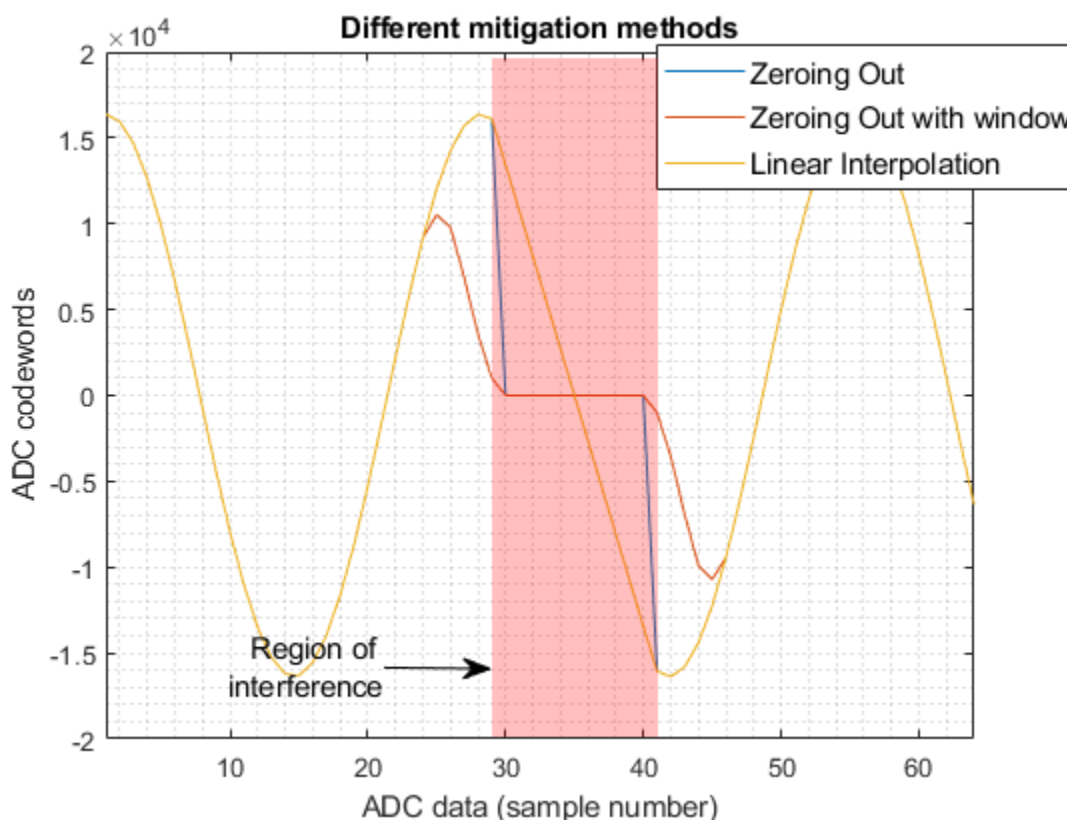


Figure 24-25. Interference Mitigation Block



**Figure 24-26. Interference Mitigation Methods**

If `INTF_MITG_PATH_SEL = 1`, then the interference mitigation block performs a windowed zero-out, where a smoothing window is applied to the edge samples of the interference-affected set of samples, in order to reduce the side-lobe increase that can happen with abrupt windowing. The windowing function that is applied is programmable via the `INTF_MITG_WINDOW_PARAM<n>` registers. The manner in which this smoothing window is applied is as follows.

Assume that the input sample array is:  $[x[0], x[1], x[2], x[3], x[4], x[5], x[6], x[7], x[8], x[9], x[10]]$ . Assume that the IIB array input is of the form  $[0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 0]$ . (In other words, the 6<sup>th</sup> and 7<sup>th</sup> samples are corrupted by interference.) Then, the output of the windowed zero-out module would be:

$[x[0]w[4], x[1]w[3], x[2]w[2], x[3]w[1], x[4]w[0], 0, 0, x[7]w[0], x[8]w[1], x[9]w[2], x[10]w[3]]$ .

Where  $w[n] = \text{INTF\_MITG\_WINDOW\_PARAM}<n>$ . Note that with `2XMODE = 1`, two different window coefficients are applied to two input samples per clock. For example  $w[4]$  and  $w[3]$  applied to  $x[0]$  and  $x[1]$ .

Note that the samples that are affected by interference are always zeroed out. The window is applied on the neighbours of these samples, so as to smoothen the transition to zero.

If `INTF_MITG_PATH_SEL = 2`, then the interference mitigation block performs a linear interpolation between the two good (i.e., non-interference-affected) samples at the start and end of the interference affected set of samples and this linear interpolation procedure is used to replace the interference affected samples themselves. The number of interference affected samples between these good samples can be any number. However, if it exceeds 32, then the 'last good sample' is pushed out in the place of affected samples until a new good sample arrives. Then linear interpolation will be performed across the 32 remaining affected samples. In case of `2XMODE = 1`, the linear interpolation is applied to both samples in a pair even if only one of them has `IIB = 1`.

### 24.2.1.6 Complex Multiplication

In addition to interference zero-out, the pre-processing block contains a complex multiplication sub-block. The purpose of this sub-block (Figure 24-27) is to enable several assorted capabilities that require complex multiplication of the input samples. The CMULT\_MODE register is used to enable and configure the complex multiplication functionality. The complex multiplication sub-block can be disabled (bypassed) by the setting CMULT\_MODE to 0b0000.

There are three modes of the complex multiplier supported as follows. They are frequency shifter mode, scalar multiplication mode and fine frequency shifter mode. In all the three modes of the complex multiplier, one only complex multiplication is performed every clock cycle. However, for real inputs with 2XMODE = 1, two real scalar multiplications per clock are possible.

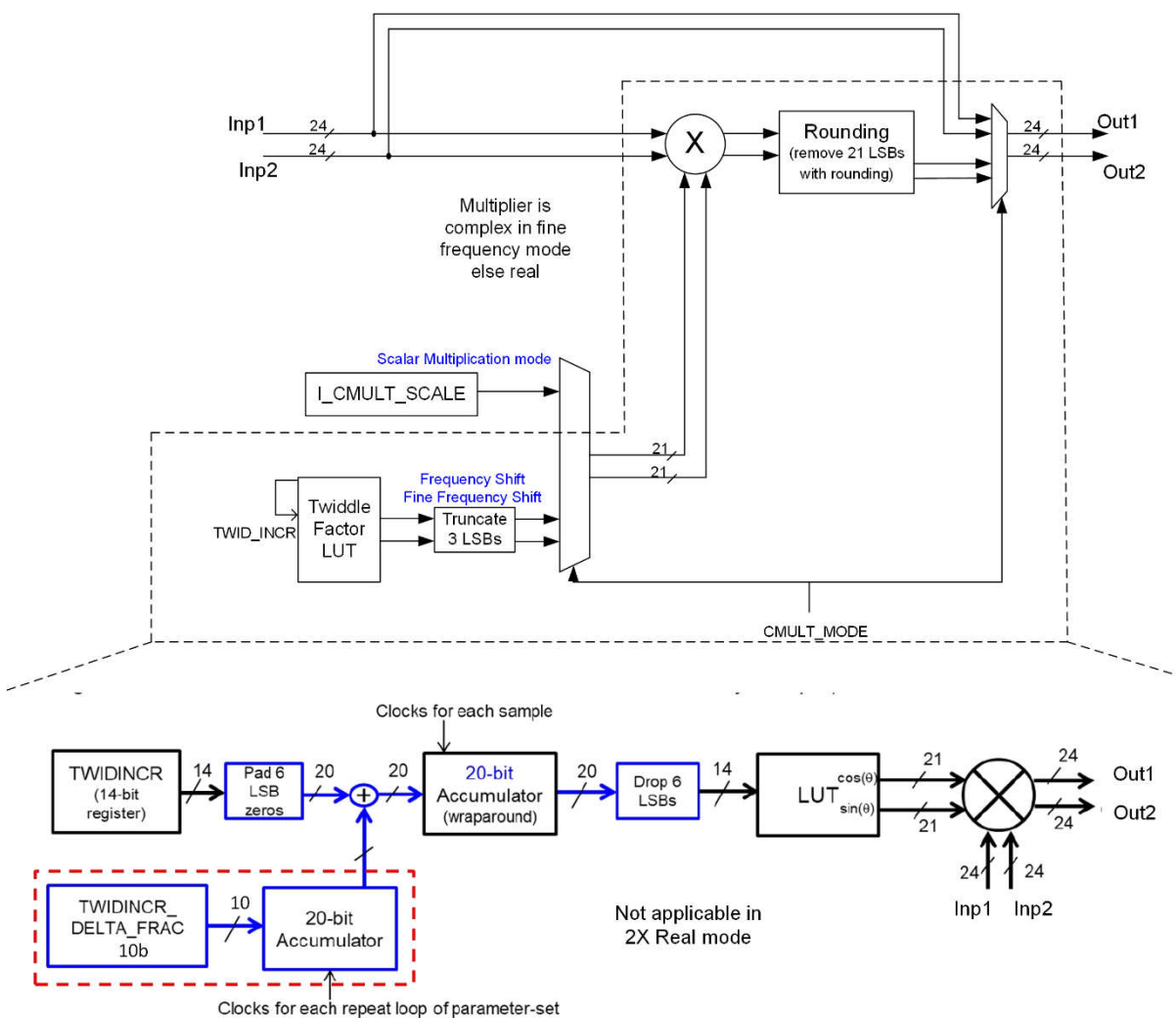


Figure 24-27. Complex Multiplication Capability in Pre-Processing Block

- Frequency shifter mode:** If the register value is CMULT\_MODE = 0001b, then the complex multiplier functions as a frequency shifter, which can be used to de-rotate the input samples by a certain frequency. This de-rotation is accomplished using cos, sin values from a twiddle factor look-up table (LUT). This LUT contains the (compressed) equivalent of the cos, sin values corresponding to the 16384 long sequence  $\exp(-j \cdot 2 \cdot \pi \cdot (0:16383)/16384)$ . TWIDINCR is used to specify the de-rotation frequency, by specifying how much the phase should change for each successive input sample (that register controls how much the LUT read index

increments every sample) as shown in (Figure 24-27). The starting phase in this mode is always zero, since the 20-bit accumulator always starts at zero for each iteration.

Note that although the figure shows another TWIDINCR\_DELTA\_FRAC register (portions shown in the red dotted box), that functionality is only applicable for CMULT\_MODE = 1010b described later and it is not applicable in the present complex multiplier mode (0001b).

Note that complex frequency shift is applicable to only complex inputs or single-sample real input. This mode is not applicable with 2XMODE = 1.

- **Scalar multiplication mode:** This mode is selected by setting CMULT\_MODE = 0101b. It supports two options – multiplication by a complex scalar that remains constant across all iterations or by a complex scalar that changes per iteration. In case of 2XMODE = 1, then a purely real scalar is applied

If the register bit CMULT\_SCALE\_EN = 0, then the complex multiplier functions in constant scalar multiplication mode. This feature is useful if the input samples need to be scaled by some constant factor. In this case, the complex multiplier will multiply each input sample with a 21-bit scalar complex number that is programmed in ICMULT\_SCALE0 and QCMULT\_SCALE0 registers (for I and Q value, each having 21 bits). The ICMULT\_SCALE0 and QCMULT\_SCALE0 registers are common registers and not part of parameter-set. For 2XMODE = 1, note that only ICMULT\_SCALE0 is applied for all samples.

To multiply the input samples for different iterations (channels) with different complex scalars, set CMULT\_SCALE\_EN = 1. In this mode, upto 4 different complex scalars are supported, viz. from ICMULT\_SCALE0, QMULT\_SCALE0 to ICMULT\_SCALE3, QMULT\_SCALE3 that are used for multiplication per iteration. TWID\_INCR Register has no implication in this mode. In case of 2XMODE = 1, then only ICMULT\_SCALE0, .. ICMULT\_SCALE3 are applied across iterations.

- **Frequency shifter mode with fine frequency increment:** This mode is enabled using CMULT\_MODE = 1010b. This mode is an extension of the Frequency shifter mode (previously explained, where CMULT\_MODE = 0001b). In this mode, in addition to the previously explained frequency shifter functionality, there is another signed 10-bit offset TWIDINCR\_DELTA\_FRAC that can be added to the de-rotation frequency, such that the de-rotation frequency changes incrementally after every “execution count”. The TWIDINCR\_DELTA\_FRAC value is automatically accumulated to the de-rotation frequency at the end of the current execution of the current parameter-set with CMULT\_MODE = 1010b. The fixed-point design of this datapath is illustrated in Figure 24-27). In effect, the input samples  $x(n)$  for  $n = 0$  to SRCACNT-1 are multiplied by the sequence:  $\exp(-j * 2\pi * ((TWIDINCR + 2^{(-6)} * TWIDINCR\_DELTA\_FRAC * execution\_count) * (0:SRCACNT-1) / 16384))$ . The execution\_count here refers to the current execution count of the parameter-set with CMULT\_MODE = 1010b. The TWIDINCR\_DELTA\_FRAC accumulator is reset only via software by writing to the TWID\_INCR\_DELTA\_FRAC\_RESET\_SW register bit. Note that the TWIDINCR\_DELTA\_FRAC register is applicable only in this mode of the complex multiplier and is ignored in all other modes. This mode is not applicable with 2XMODE = 1.

#### 24.2.1.7 Pre-Processing Block – Register Descriptions

Table 24-8 lists all the registers of the pre-processing block. As explained in Section 24.1.1.1, some of the registers are common (common for all parameter-sets) registers, whereas, some others are “part of each parameter-set”. For each register, this distinction is captured as part of the register description in Table 24-8.

**Table 24-8. Pre-Processing Registers**

Register.field	Width	Parameter-Set? (Y/N)	Description
DCEST_RESET_MODE	2	Y	2-bit field that controls the reset behavior for all 12 DC accumulators 00 : Hold Accumulator state without updating 01 : Reserved 10 : Reset at start of param-set (i.e., per-chirp DC estimation). 11 : Reset at start of param-set only if loop counter is 0 (i.e., per-frame DC estimation)

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
DC_EST_CTRL.dc_est_scale	9	N	Programmable fine scaling for DC estimation: 9-bit scale applied to all 12 DC accumulators. This is followed by right shift and truncation. Multiplies the accumulator output by DC_EST_SCALE/256. Default value is 256 giving a scale of 1.0. Setting it to 128, gives a scale of 0.5.
DC_EST_CTRL.dc_est_shift	4	N	Programmable right shift for DC estimation: Right bit-shift applied to all 12 DC accumulator outputs. Cannot be bypassed. Accumulator outputs are scaled by $2^{(8 + 6 + \text{DC\_EST\_SHIFT})}$ . Valid range for this register is 0 to 14 (i.e., scaling of $2^{14}$ to $2^{28}$ ). Note that DC_EST_SHIFT = 15 is not supported.
DC_ACC_I_<n>_VAL_LSB. Dc_acc_i_<n>_val_lsb n=0,1,..11	32	N	These read-only registers provide the lower 32 bits of 36b DC estimation accumulator values –I&Q for 12 streams for processor read-out.
DC_ACC_I_<n>_VAL_MSB. Dc_acc_i_<n>_val_msb n=0,1,..11	4	N	These read-only registers provide the upper 4 bits of 36b DC estimation accumulator values –I&Q for 12 streams for processor read-out.
DC_EST_RESET_SW.dc_est_reset_sw	1	N	Software reset for DC accumulators: Setting this register bit to 1 resets all 12 DC estimation accumulators. This is a self-clearing reset bit.
DC_EST_I_<n>_VAL.dc_est_i_<n>_val DC_EST_Q_<n>_VAL.dc_est_q_<n>_val n=0,1,..11	24	N	These read-only registers provide the DC estimates – I&Q for 12 streams – for the processor to read.
DC_ACC_CLIP_STATUS.dc_acc_clip_status	12	N	Clip status indication (read-register) for the 12 DC accumulators (both I and Q combined). Value of 1 indicates a clipping event occurred.
DC_EST_CLIP_STATUS.dc_est_clip_status	12	N	Clip status indication (read-register) for the 12 DC estimates (both I and Q combined). Value of 1 indicates a clipping event occurred.
DCSUB_EN	1	Y	Enable or Disable DC subtraction. If this register bit is set to 1, DC subtraction is enabled. Else, it is disabled.
DCSUB_SELECT	1	Y	Source select for DC subtraction: 0 : Value comes from processor via DC_SW_I_<n> & DC_SW_Q_<n> 1: Value comes from built-in DC estimation hardware, i.e., DCEST_I_<n> & DCEST_Q_<n>
DC_I_<n>_SW.dc_i_<n>_sw DC_Q_<n>_SW.dc_q_<n>_sw n=0,1,..11	24	N	User-programmed DC values used for DC subtraction. These registers are relevant only when DCSUB_SELECT is 0.
DC_SUB_CLIP.dc_sub_clip	1	N	Clip status indication (read-register) for DC subtraction node (both I and Q combined). Value of 1 indicates a clipping event occurred.
INTF_LOC_THRESH_EN	1	Y	Enable/Disable for Interference localization (marking out): This registerbit controls the enable/disable for the interference marking (setting Interference Indicator Bit) feature. The feature is enabled if this register bit is set to 1.



**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_LOC_THRESH_MAG<n>_SW.intf_loc_thresh_mag<n>_sw n=0..11	24	N	Software Interference threshold for Magnitude These registers are used to specify the user-programmed threshold for marking out samples affected by interference in the Interference localization block. The magnitude of each incoming samples is compared with this threshold to decide whether it is corrupted by interference or not.
INTF_LOC_THRESH_MAGD IFF<n>_SW.intf_loc_thresh_magdiff<n>_sw n=0...11	24	N	Software Interference threshold for Magnitude of backward difference These registers are used to specify the user-programmed threshold for marking out samples affected by interference in the Interference localization block. The magnitude of backward difference of incoming samples is compared with this threshold to decide whether it is corrupted by interference or not.
INTF_LOC_THRESH_MODE	2	Y	Interference detection mode selection: This register is used to control the mode for interference detection in the Interference localization block. 00 : Magnitude OR Magnitude difference 01: Only Magnitude difference 10: Only Magnitude 11 : Magnitude AND Magnitude difference
INTF_LOC_THRESH_SEL	2	Y	Select the source of interference threshold 0 : User-defined threshold via INTERFTHRESH_MAG_SW and INTERFTHRESH_MAGDIFF_SW 1 : Single threshold based on built-in interference statistics outputs using sum value across collected interference statistics 2 : Threshold based on built-in interference statistics outputs, with each statistic being used for corresponding iteration (RX channel)
INTF_STATS_RESET_MODE	2	Y	Reset mode control for Interference statistics accumulators: Controls the reset behavior for all 12 magnitude and magdiff accumulators. 00 : Hold Accumulator state without updating 01 : Free-running accumulator mode 10 : Reset at start of parameter-set (i.e., per-chirp accumulation). 11 : Reset at start of parameter-set only if loop counter is 0 (i.e., per-frame)
INTF_STATS_CTRL.intf_stats_mag_scale	8	N	Programmable fine scaling for Interference statistics Magnitude: Scaling applied to INTF_STATS_MAGACC<n> from interference statistics block.
INTF_STATS_CTRL.intf_stats_mag_shift	3	N	Programmable right shift for Interference statistics Magnitude: Right bit-shift applied to the interference magnitude accumulator. Total right shift of the accumulator is $2^{(3+6+INTF\_STATS\_MAG\_SHIFT)}$ . Valid range for this register is 0 to 6 (i.e., the total right shift can't be more than $2^{15}$ ).
INTF_STATS_CTRL.intf_stats_magdiff_scale	8	N	Programmable fine scaling for Interference statistics MagDiff: Scaling applied to INTF_STATS_MAGDIFFACC<n> from interference statistics block.



**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_STATS_CTRL.intf_stats_magdiff_shift	3	N	Programmable right shift for Interference statistics MagDiff. Right bit-shift applied to the interference magdiff accumulator. Total right shift of the accumulator is $2^{(3+6+INTF\_STATS\_MAGDIFF\_SHIFT)}$ . Valid range for this register is 0 to 6 (i.e., the right shift can't be more than $2^{15}$ ).
INTF_STATS_MAG_ACC<n>_LSB.intf_stats_mag_acc<n>_lsb	32	N	These read-only registers provide the lower 32 bits of 36b magnitude accumulator values –I&Q 12 streams for processor read-out.
INTF_STATS_MAG_ACC<n>_MSB.intf_stats_mag_acc<n>_msb	4	N	These read-only registers provide the upper 4 bits of 36b magnitude accumulator values –I&Q 12 streams for processor read-out.
INTF_STATS_MAGDIFF_ACC<n>_LSB.intf_stats_magdiff_acc<n>_lsb	32	N	These read-only registers provide the lower 32 bits of 36b magnitude difference accumulator values –I&Q 12 streams for processor read-out.
INTF_STATS_MAGDIFF_ACC<n>_MSB.intf_stats_magdiff_acc<n>_msb	4	N	These read-only registers provide the upper 4 bits of 36b magnitude difference accumulator values –I&Q 12 streams for processor read-out.
INTF_STATS_RESET_SW.intf_stats_reset_sw	1	N	Software reset bit for all the interference statistics accumulators. This is a self-clearing reset bit.
INTF_LOC_THRESH_MAG<n>_VAL.intf_loc_thresh_mag<n>_val	24	N	Read-only thresholds – scaled and shifted INTF_STATS_MAGACC<n> of interference statistics block
INTF_STATS_SUM_MAG_VAL.intf_stats_sum_mag_val	24	N	Sum of INTF_LOC_THRESH_MAG<n>_VAL, based on number of iterations. Useful as single magnitude threshold value across all iterations
INTF_LOC_THRESH_MAGDIFF<n>_VAL.intf_loc_thresh_magdiff<n>_val	24	N	Read-only thresholds – scaled and shifted INTF_STATS_MAGACCDIFF<n> of interference statistics block
INTF_STATS_SUM_MAGDIFF_VAL.intf_stats_sum_magdiff_val	24	N	Sum of INTF_LOC_THRESH_MAGDIFF<n>_VAL, based on number of iterations. Useful as single magnitude difference threshold value across all iterations
INTF_STATS_SUM_MAG_VAL_CLIP_STATUS.intf_stats_sum_mag_val_clip_status	12	N	Read-only clip status indication register for 12 interference statistics magnitude accumulators INTF_STATS_MAGACC<n>. Value of 1 indicates a clipping event occurred in atleast one accumulator
INTF_STATS_SUM_MAGDIFF_VAL_CLIP_STATUS.intf_stats_sum_magdiff_val_clip_status	12	N	Read-only clip status indication register for 12 interference statistics magnitude-difference accumulators INTF_STATS_MAGACCDIFF<n>. Value of 1 indicates a clipping event occurred in atleast one accumulator
INTF_STATS_ACC_CLIP_STATUS.intf_stats_mag_accumulator_clip_status	12	N	Read-only clip status indication register for 12 magnitude based interference threshold. INTF_LOC_THRESH_MAG<n>_VAL. Value of 1 indicates a clipping event occurred in atleast one estimate
INTF_STATS_ACC_CLIP_STATUS.intf_stats_magdiff_accumulator_clip_status	12	N	Read-only clip status indication register for magnitude-difference based interference threshold. INTF_LOC_THRESH_MAGDIFF<n>_VAL. Value of 1 indicates a clipping event occurred in atleast one estimate
INTF_STATS_THRESH_CLIP_STATUS.intf_stats_thresh_mag_clip_status	1	N	Read-only clip status for sum of all magnitude thresholds computed by the statistics block. Value of 1 indicates that the sum clipped.
INTF_STATS_THRESH_CLIP_STATUS.intf_stats_thresh_magdiff_clip_status	1	N	Read-only clip status for sum of all magnitude difference thresholds computed by statistics block. Value of 1 indicates that the sum clipped.
INTF_LOC_COUNT_ALL_CHIRP.intf_loc_count_all_chirp	12	N	Read-only register indicating the number of samples that exceeded the threshold in a given param-set. The count is saturated to $2^{12} - 1$ .

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_LOC_COUNT_ALL_FRAME.intf_loc_count_all_frame	20	N	Read-only register indicating the number of samples that exceeded the threshold across multiple executions of same param-set. The count is saturated to $2^{20} - 1$ .
INTF_MITG_EN	1	Y	If this bit is set, the interference mitigation path is activated, else it is bypassed.
INTF_MITG_PATH_SEL	2	Y	Based on the value of this register, one of the three paths is activated. 00b: Simple Zeroing out 01b: Windowed Zeroing out 10b: Linear Interpolation 11b: Reserved
INTF_MITG_WINDOW_PARAM_<n>.intf_mitg_window_param_<n> n=0..4	5 (each)	N	This is a programmable array of window parameters. Each window parameter is an unsigned 5 bit integer. The length of the array is 5. The parameters of the window are assumed to be monotonically ascending. For example : $val = \text{floor}(\text{hanning}(14) * 32)$ $\text{INTF\_MITG\_WINDOW\_PARAM} = \text{val}(2:6);$ If a shorter window (of length less than 5) is desired, some of the earlier window parameters can be set to 31. This sets the window parameter to 31/32 (or ~1).
INTF_MITG_CNTTHRESH	5	Y	The (total) number of non-zero IIB within the 'Hysteresis window' should exceed this threshold for the sample-under-test to be considered to be affected by interference. Range : 0 to 31.
INTF_MITG_RIGHT_HYST_ORD	4	Y	The length of the IIB array considered on the right side of (i.e. after) the sample under test. Range : 0 to 15.
INTF_MITG_LEFT_HYST_ORD	4	Y	The length of the IIB array considered on the left side of (i.e. before) the sample under test. Range : 0 to 15.
CMULT_MODE	4	Y	Complex multiplication mode selection: This register is used to configure the mode of the complex multiplication sub-block. A value of 0000b disables/bypasses the complex multiplication. Any other value chooses one of nine available modes of operation. Detailed description of the nine modes in the main description section.

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
CMULT_SCALE_EN	1	Y	<p>Complex multiplier iteration enable :</p> <p>This register bit is applicable in certain modes of the complex multiplication pre-processing to enable per-iteration change of the complex scalar coefficient.</p> <p>When using scalar multiplication mode of the complex multiplier (CMULT_MODE = 0101b), if CMULT_SCALE_EN is set to 1, then the input samples are multiplied by a different complex scalar (i.e., ICMULTSCALE0, QMULTSCALE0 to ICMULTSCALE11, QMULTSCALE11) for each iteration. Else, a constant complex scalar ICMULTSCALE0 and QCMULTSCALE0 is applied to all samples across all iterations.</p> <p>When using vector multiplication mode (CMULT_MODE = 0110b), if CMULT_SCALE_EN is set to 1, then instead of pulling the coefficients for vector multiplication from the Vector Multiplication Coefficients RAM, the input samples are multiplied successively by ICMULTSCALE0, QMULTSCALE0 to ICMULTSCALE11, QMULTSCALE11 registers.</p>
ICMULT_SCALE<n>.icmult_scale<n> QCMULT_SCALE<n>.qcmult_scale<n> n = 0 ..11	21	N	<p>Coefficients for Complex multiplication:</p> <p>Refer the description for CMULT_SCALE_EN register.</p>
VEC_MULT_RAM[1024]	32	N	<p>Vector multiplication RAM : Stores the complex vector multiplication coefficients used in modes 6, 7 and 8. Layout shown in Fig. 28-34</p>
TWIDINCR	14	Y	<p>Frequency shifter configuration:</p> <p>When the complex multiplication sub-block is programmed in one of the frequency shifter modes (CMULT_MODE = 0001b or 0010b), this register is used to indicate the amount of frequency shift.</p> <p>When the complex multiplication sub-block is programmed in FFT stitching mode (CMULT_MODE = 0011b), the last two bits of this register specify whether it is 4K or 8K FFT stitching. Specifically, if the last two bits are 01b, then it is 4K FFT stitching and if the last two bits are 10b, then it is 8K FFT stitching. Values of 00b and 11b are reserved. Also, the 12 MSB bits of this register must be kept zero in the FFT stitching mode.</p> <p>In all other modes of the complex multiplication sub-block, this 14-bit register must be kept as 0.</p> <p>When the complex multiplication sub-block is programmed with CMULT_MODE = 0110b, 0111b or 1000b, then the 12 MSBs of this register are used as an address offset for the Vector Multiplication Coefficients RAM (the 2 LSBs must be kept 0).</p> <p>When the complex multiplication sub-block is programmed with CMULT_MODE = 1001b, then the 6 LSBs of this register are used as an address offset for TWID_ANGLE_RAM, with bit 13 enabling auto-address increment over iterations and bit 12 enabling address saturation or address roll-over after 63.</p>
TWID_INCR_DELTA_FRAC.twid_incr_delta_frac	10	N	<p>Fractional frequency increment per execution of the parameter-set:</p> <p>Frequency shift value to be accumulated at the end of current parameter-set. Refer main description for more details.</p>
DSS_HWA_DEROT_RAM	32	N	

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
RECWIN_MODE	1	Y	Recursive window mode select bit.(0) –the K value increments with iteration. (1) –the K value increments with paramset execution count. K always starts from 0
RECWIN_RESET_SW.recwin_reset_sw	1	N	Software reset bit for recursive window K value. This is a self-clearing reset bit.
RECWIN_INIT_KVAL.recwin_init_kval	12	N	RESERVED. This is a reserved register and should be always kept 0.
TWID_INCR_DELTA_FRAC_RESET_SW.twid_incr_delta_frac_reset_sw	1	N	Software reset bit for fine frequency increment accumulator (in CMULT_MODE = 1010b). This is a self-clearing reset bit.
TWID_INCR_DELTA_FRAC_CLIP_STATUS.twid_incr_delta_frac_clip_status	1	N	Read-only register bit that indicates clip status for the fine-frequency increment accumulator (in CMULT_MODE = 1010b).
BPM_EN	1	Y	Enable/Disable BPM removal: This registerbit specifies whether the BPM removal needs to be enabled or not. If this register is set, then BPM removal is enabled prior to feeding samples from the input formatter into the core computational unit.
BPM_PATTERN<n>.bpm_pattern_<n> n=0..7	256	N	BPM pattern: Specifies the BPM pattern to be used to multiply the input samples if BPM removal is enabled. The 256-bit word is split into 8 32-bit words as [BPM_PATTERN_7, BPM_PATTERN_6, BPM_PATTERN_5, BPM_PATTERN_4, BPM_PATTERN_3, BPM_PATTERN_2, BPM_PATTERN_1, BPM_PATTERN_0]
BPM_RATE.bpm_rate	10	N	BPM rate: Specifies the number of input samples corresponding to each BPM bit. Minimum valid value for this register is 1.
BPM_PHASE	4	Y	BPM starting phase: Specifies the starting phase of the BPM pattern periodicity. For more information, see the detailed description.
CHAN_COMB_EN	1	Y	Enable/Disable channel combining: If this register bit is set to 1, then the channel combining feature is enabled.
CHAN_COMB_VEC_<n>.chan_comb_vec_<n> n=0..7	256	N	Sample index indicator for Channel combining : This register indicates the sample indices that need to be combined in the Channel combiner block. A '01' or '10' transition demarcates the groups. The 256-bit word is split into 8 32-bit words as [CHAN_COMB_VEC_7, CHAN_COMB_VEC_6, CHAN_COMB_VEC_5, CHAN_COMB_VEC_4, CHAN_COMB_VEC_3, CHAN_COMB_VEC_2, CHAN_COMB_VEC_1, CHAN_COMB_VEC_0]
CHAN_COMB_SIZE.chan_comb_size	8	N	This register indicates the number of samples after channel combination in each iteration.
CHANNEL_COMB_CLIP_STATUS.channel_comb_clip_status	1	N	Clip status indication (read-register) during channel combining. Value of 1 indicates a clipping event occurred.
ZERO_INSERT_EN	1	Y	Enable/Disable zero-insertion: If this register bit is set to 1, then the zero-insertion feature is enabled.

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
ZERO_INSERT_NUM.zero_insert_num		N	This register indicates the number of zeros to be inserted in each iteration.
DCEST_RESET_MODE	2	Y	2-bit field that controls the reset behavior for all 8 DC accumulators b'00 : Hold Accumulator state without updating b'01 : Reserved b'10 : Reset at start of param-set (i.e., per-chirp DC estimation). b'11 : Reset at start of param-set only if loop counter is 0 (i.e., per-frame DC estimation)
DCEST_PROFILE_SELECT	1	Y	DC estimation profile selection bit to choose which set of shift/scale settings to be applied and which set of estimation registers be updated b'0 : Profile 0, DC_EST_I_<0..3>.val b'1 : Profile 1, DC_EST_I_<4..7>.val
DC_EST_CTRL.dc_est_scale_profile0	9	N	Programmable fine scaling for DC estimation in Profile 0: 9-bit scale applied to 4 DC accumulator values in profile0. This is followed by right shift and truncation. Multiplies the accumulator output by DC_EST_SCALE/256. Default value is 256 giving a scale of 1.0. Setting it to 128, gives a scale of 0.5.
DC_EST_CTRL.dc_est_scale_profile1	9	N	Programmable fine scaling for DC estimation in Profile 1
DC_EST_CTRL.dc_est_shift_profile0	4	N	Programmable right shift for DC estimation in Profile0: Right bit-shift applied to all 4 DC accumulator outputs in profile0. Cannot be bypassed. Accumulator outputs are scaled by $2^{(8 + 6 + \text{DC\_EST\_SHIFT})}$ . Valid range for this register is 0 to 14 (i.e., scaling of $2^{14}$ to $2^{28}$ ). Note that DC_EST_SHIFT = 15 is not supported.
DC_EST_CTRL.dc_est_shift_profile1	4	N	Programmable right shift for DC estimation in Profile1
DC_ACC_I_<n>_VAL_LSB. dc_acc_i_<n>_val_lsb n=0,1,..7	32	N	These read-only registers provide the lower 32 bits of 36b DC estimation accumulator values –I&Q for 8 streams for processor read-out.
DC_ACC_I_<n>_VAL_MSB. dc_acc_i_<n>_val_msb n=0,1,..7	4	N	These read-only registers provide the upper 4 bits of 36b DC estimation accumulator values –I&Q for 8 streams for processor read-out.
DC_EST_RESET_SW.dc_est_reset_sw	1	N	Software reset for DC accumulators: Setting this register bit to 1 resets all 8 DC estimation accumulators. This is a self-clearing reset bit.
DC_EST_I_<n>_VAL.dc_est_i_<n>_val DC_EST_Q_<n>_VAL.dc_est_q_<n>_val n=0,1,..7	24	N	These read-only registers provide the DC estimates – I&Q for 8 streams – for the processor to read. Only I-values updated with 2XMODE = 1
DC_ACC_CLIP_STATUS.dc_acc_clip_status	8	N	Clip status indication (read-register) for the 8 DC accumulators (both I and Q combined). Value of 1 indicates a clipping event occurred.
DC_EST_CLIP_STATUS.dc_est_clip_status	8	N	Clip status indication (read-register) for the 8 DC estimates (both I and Q combined). Value of 1 indicates a clipping event occurred.
DCSUB_EN	1	Y	Enable or Disable DC subtraction. If this register bit is set to 1, DC subtraction is enabled. Else, it is disabled.

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
DCSUB_SELECT	1	Y	Source select for DC subtraction: 0: Value comes from processor via DC_SW_I<n> & DC_SW_Q<n> 1: Value comes from built-in DC estimation hardware, i.e., DCEST_I<n> & DCEST_Q<n> n=0..3 in Profile0, n=4..7 in Profile1
DC_I<n>_SW.dc_i<n>_sw DC_Q<n>_SW.dc_q<n>_sw n=0,1,..7	24	N	User-programmed DC values used for DC subtraction. These registers are relevant only when DCSUB_SELECT is 0.
DC_SUB_CLIP.dc_sub_clip	1	N	Clip status indication (read-register) for DC subtraction node (both I and Q combined). Value of 1 indicates a clipping event occurred.
INTF_LOC_THRESH_EN	1	Y	Enable/Disable for Interference localization (marking out): This registerbit controls the enable/disable for the interference marking (setting Interference Indicator Bit) feature. The feature is enabled if this register bit is set to 1.
INTF_LOC_THRESH_MAG<n>_SW.intf_loc_thresh_mag <n>_sw n=0..3	24	N	Software Interference threshold for Magnitude These registersare used to specify the user-programmed threshold for marking out samples affected by interference in the Interference localization block. The magnitude of each incoming samples is compared with this threshold to decide whether it is corrupted by interference or not.
INTF_LOC_THRESH_MAGD IFF<n>_SW.intf_loc_thresh_magdiff<n>_sw n=0...3	24	N	Software Interference threshold for Magnitude of backward difference These registersare used to specify the user-programmed threshold for marking out samples affected by interference in the Interference localization block. The magnitude of backward difference of incoming samples is compared with this threshold to decide whether it is corrupted by interference or not.
INTF_LOC_THRESH_MODE	2	Y	Interference detection mode selection: This register is used to control the mode for interference detection in the Interference localization block. 00 : Magnitude OR Magnitude difference 01: Only Magnitude difference 10: Only Magnitude 11 : Magnitude AND Magnitude difference
INTF_LOC_THRESH_SEL	2	Y	Select the source of interference threshold 0 : User-defined threshold via INTERFTHRESH_MAG_SW and INTERFTHRESH_MAGDIFF_SW 1 : Single threshold based on built-in interference statistics outputs using sum value across collected interference statistics 2 : Threshold based on built-in interference statistics outputs, with each statistic being used for corresponding iteration (RX channel)
INTF_STATS_RESET_MODE	2	Y	Reset mode control for Interference statistics accumulators: Controls the reset behavior for all 4 magnitude and magdiff accumulators. 00 : Hold Accumulator state without updating 01 : Free-running accumulator mode 10 : Reset at start of parameter-set (i.e.,per-chirp accumulation). 11 : Reset at start of parameter-set only if loop counter is 0 (i.e., per-frame)

**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_STATS_CTRL.intf_stats_mag_scale	8	N	Programmable fine scaling for Interference statistics Magnitude: Scaling applied to INTF_STATS_MAGACC<n> from interference statistics block.
INTF_STATS_CTRL.intf_stats_mag_shift	3	N	Programmable right shift for Interference statistics Magnitude: Right bit-shift applied to the interference magnitude accumulator. Total right shift of the accumulator is $2^{(3+6+INTF\_STATS\_MAG\_SHIFT)}$ . Valid range for this register is 0 to 6 (i.e., the total right shift can't be more than $2^{15}$ ).
INTF_STATS_CTRL.intf_stats_magdiff_scale	8	N	Programmable fine scaling for Interference statistics MagDiff: Scaling applied to INTF_STATS_MAGDIFFACC<n> from interference statistics block.
INTF_STATS_CTRL.intf_stats_magdiff_shift	3	N	Programmable right shift for Interference statistics MagDiff: Right bit-shift applied to the interference magdiff accumulator. Total right shift of the accumulator is $2^{(3+6+INTF\_STATS\_MAGDIFF\_SHIFT)}$ . Valid range for this register is 0 to 6 (i.e., the right shift can't be more than $2^{15}$ ).
INTF_STATS_MAG_ACC<n>_LSB.intf_stats_mag_acc<n>_lsb. n= 0..3	32	N	These read-only registers provide the lower 32 bits of 36b magnitude accumulator values – 4 streams for processor read-out.
INTF_STATS_MAG_ACC<n>_MSB.intf_stats_mag_acc<n>_msb. n= 0..3	4	N	These read-only registers provide the upper 4 bits of 36b magnitude accumulator values – 4 streams for processor read-out.
INTF_STATS_MAGDIFF_AC<n>_LSB.intf_stats_magdiff_acc<n>_lsb. n= 0..3	32	N	These read-only registers provide the lower 32 bits of 36b magnitude difference accumulator values – 4 streams for processor read-out.
INTF_STATS_MAGDIFF_AC<n>_MSB.intf_stats_magdiff_acc<n>_msb. n= 0..3	4	N	These read-only registers provide the upper 4 bits of 36b magnitude difference accumulator values – 4 streams for processor read-out.
INTF_STATS_RESET_SW.intf_stats_reset_sw	1	N	Software reset bit for all the interference statistics accumulators. This is a self-clearing reset bit.
INTF_LOC_THRESH_MAG<n>_VAL.intf_loc_thresh_mag<n>_val. n= 0..3	24	N	Read-only thresholds – scaled and shifted INTF_STATS_MAGACC<n> of interference statistics block
INTF_STATS_SUM_MAG_VAL.intf_stats_sum_mag_val. n= 0..3	24	N	Sum of INTF_LOC_THRESH_MAG<n>_VAL, based on number of iterations. Useful as single magnitude threshold value across all iterations
INTF_LOC_THRESH_MAGDIFF<n>_VAL.intf_loc_thresh_magdiff<n>_val. n= 0..3	24	N	Read-only thresholds – scaled and shifted INTF_STATS_MAGACCDIFF<n> of interference statistics block
INTF_STATS_SUM_MAGDIFF_VAL.intf_stats_sum_magdiff_val. n= 0..3	24	N	Sum of INTF_LOC_THRESH_MAGDIFF<n>_VAL, based on number of iterations. Useful as single magnitude difference threshold value across all iterations
INTF_STATS_SUM_MAG_CLIP_STATUS.intf_stats_sum_mag_val_clip_status	4	N	Read-only clip status indication register for 12 interference statistics magnitude accumulators INTF_STATS_MAGACC<n>. Value of 1 indicates a clipping event occurred in atleast one accumulator



**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
INTF_STATS_SUM_MAGDIFF_VAL_CLIP_STATUS.intf_stats_sum_magdiff_val_clip_status	4	N	Read-only clip status indication register for 12 interference statistics magnitude-difference accumulators INTF_STATS_MAGACCDIFF<n>. Value of 1 indicates a clipping event occurred in atleast one accumulator
INTF_STATS_ACC_CLIP_STATUS.intf_stats_mag_accumulator_clip_status	4	N	Read-only clip status indication register for 12 magnitude based interference threshold. INTF_LOC_THRESH_MAG<n>_VAL. Value of 1 indicates a clipping event occurred in atleast one estimate
INTF_STATS_ACC_CLIP_STATUS.intf_stats_magdiff_accumulator_clip_status	4	N	Read-only clip status indication register for magnitude-difference based interference threshold. INTF_LOC_THRESH_MAGDIFF<n>_VAL. Value of 1 indicates a clipping event occurred in atleast one estimate
INTF_STATS_THRESH_CLIP_STATUS.intf_stats_thresh_mag_clip_status	1	N	Read-only clip status for sum of all magnitude thresholds computed by the statistics block. Value of 1 indicates that the sum clipped.
INTF_STATS_THRESH_CLIP_STATUS.intf_stats_thresh_magdiff_clip_status	1	N	Read-only clip status for sum of all magnitude difference thresholds computed by statistics block. Value of 1 indicates that the sum clipped.
INTF_LOC_COUNT_ALL_CHIRP.intf_loc_count_all_chirp	12	N	Read-only register indicating the number of samples that exceeded the threshold in a given param-set. The count is saturated to $2^{12} - 1$ .
INTF_LOC_COUNT_ALL_FRAME.intf_loc_count_all_frame	20	N	Read-only register indicating the number of samples that exceeded the threshold across multiple executions of same param-set. The count is saturated to $2^{20} - 1$ .
INTF_MITG_EN	1	Y	If this bit is set, the interference mitigation path is activated, else it is bypassed.
INTF_MITG_PATH_SEL	2	Y	Based on the value of this register, one of the three paths is activated. 00b: Simple Zeroing out 01b: Windowed Zeroing out 10b: Linear Interpolation 11b: Reserved
INTF_MITG_WINDOW_PARAM<n>.intf_mitg_window_param<n> n=0..4	5 (each)	N	This is a programmable array of window parameters. Each window parameter is an unsigned 5 bit integer. The length of the array is 5. The parameters of the window are assumed to be monotonically ascending. For example : $val = \text{floor}(\text{hanning}(14) * 32)$ $\text{INTF\_MITG\_WINDOW\_PARAM} = \text{val}(2:6);$ If a shorter window (of length less than 5) is desired, some of the earlier window parameters can be set to 31. This sets the window parameter to 31/32 (or ~1).
INTF_MITG_CNTTHRESH	5	Y	The (total) number of non-zero IIB within the 'Hysteresis window' should exceed this threshold for the sample-under-test to be considered to be affected by interference. Range : 0 to 31.
INTF_MITG_RIGHT_HYST_ORD	4	Y	The length of the IIB array considered on the right side of (i.e. after) the sample under test. Range : 0 to 15.
INTF_MITG_LEFT_HYST_ORD	4	Y	The length of the IIB array considered on the left side of (i.e. before) the sample under test. Range : 0 to 15.



**Table 24-8. Pre-Processing Registers (continued)**

Register.field	Width	Parameter-Set? (Y/N)	Description
CMULT_MODE	4	Y	Complex multiplication mode selection: This register is used to configure the mode of the complex multiplication sub-block. A value of 0000b disables/bypasses the complex multiplication. Any other value chooses one of three available modes of operation. Detailed description of the three modes in the main description section.
CMULT_SCALE_EN	1	Y	Complex multiplier iteration enable : This register bit is applicable in certain modes of the complex multiplication pre-processing to enable per-iteration change of the complex scalar coefficient. When using scalar multiplication mode of the complex multiplier (CMULT_MODE = 0101b), if CMULT_SCALE_EN is set to 1, then the input samples are multiplied by a different complex scalar (i.e., ICMULTSCALE0, QMULTSCALE0 to ICMULTSCALE11, QMULTSCALE11) for each iteration. Else, a constant complex scalar ICMULTSCALE0 and QCMULTSCALE0 is applied to all samples across all iterations. Only ICMULTSCALE<n> is used when 2XMODE = 1
ICMULT_SCALE<n>.icmult_scale<n> QCMULT_SCALE<n>.qcmult_scale<n> n = 0 ..3	21	N	Coefficients for Complex multiplication: Refer the description for CMULT_SCALE_EN register.
TWIDINCR	14	Y	Frequency shifter configuration: When the complex multiplication sub-block is programmed in one of the frequency shifter modes (CMULT_MODE = 0001b or 0010b), this register is used to indicate the amount of frequency shift. In all other modes of the complex multiplication sub-block, this 14-bit register must be kept as 0.
TWID_INCR_DELTA_FRAC.twid_incr_delta_frac	10	N	Fractional frequency increment per execution of the parameter-set: Frequency shift value to be accumulated at the end of current parameter-set. Refer main description for more details.
TWID_INCR_DELTA_FRAC_RESET_SW.twid_incr_delta_frac_reset_sw	1	N	Software reset bit for fine frequency increment accumulator (in CMULT_MODE = 1010b). This is a self-clearing reset bit.
TWID_INCR_DELTA_FRAC_CLIP_STATUS.twid_incr_delta_frac_clip_status	1	N	Read-only register bit that indicates clip status for the fine-frequency increment accumulator (in CMULT_MODE = 1010b).

The accelerator includes a Compression Engine, which can compress data in order to reduce storage RAM size requirements. For example, after range dimension FFT of the received RX data, the FFT results can be input to the Compression Engine and its output can be stored in compressed form in a relatively smaller amount of RAM in the device (e.g. L3 RAM). The features and configurability of the Compression Engine are described in Part 3 of user's guide.

### 24.3 Radar Hardware Accelerator 1.5 User Guide – Part 3: Compression Engine

The radar hardware accelerator user's guide is presented in three parts. Together, these documents describe the radar hardware accelerator architecture, features, operation of various blocks and their register descriptions. The

purpose is to enable the user to understand the capabilities offered by the radar hardware accelerator and to program it appropriately to achieve the desired functionality.

The user's guide is split into three parts:

- The first part of the user guide (a separate document) provides an overview of the overall architecture and features available in the radar hardware accelerator. The main features, such as, windowing and FFT covered in the first part.
- The second part of the user guide (a separate document) covers additional features like DC Estimation and correction, Interference localization and mitigation, complex multiplication and interference statistics. This part of the user's guide assumes that the user has already read and understood Part 1 (see Radar Hardware Accelerator 1.5 User's Guide – Part 1).
- The third part of the user guide (this document) covers the compression engine.

This document is organized as follows:

- Section 1 provides an overview of the features of the compression engine.
- Section 2 describes the compression algorithms available in the engine.
- Section 3 covers information related to configuration and register descriptions.

### 24.3.1 Compression/Decompression Engine Overview

The HWA 1.5 supports only compression unlike the previous HWA 2.1. It consists of only a compression engine which takes a fixed number of samples and returns a 'block of bits' such that the block's size (in bits occupied) is a fraction of the size of the input samples.

The features of the compression engine are

1. The compression engine is designed to achieve *arbitrary* compression ratio. ('compression ratio' is defined as the ratio of 'average bit-width per sample' after compression and 'the original bit-width' before compression. In other words, a 33 % compression-ratio, results in the average bitwidth after compression being 1/3<sup>rd</sup> of the bitwidth before compression.)
2. It has two configurable algorithms for compression.
  - Block Floating Point (BFP) .
  - Exponential-Golomb Encoding (EGE).
3. It implements a 'block' based compression scheme - i.e. it takes a fixed number of samples (called a block) and creates a 'compressed block of bits' of fixed size. During the Doppler processing operation, when radar data has to be accessed or written in transpose, having each block as a fixed size simplifies the EDMA programming. The EDMA can simply access a full block (across Doppler) in much the same manner as a single range gate.
4. It is a part of the HWA as one of the programmable 'paths' in the accelerator (in addition to the FFT). It can therefore use existing capabilities/resources of the HWA (input/output formatters, state machine, looping, etc).

Note: Low compression ratios can result in 'high quantization noise'. Designers should select the appropriate 'compression ratio' after confirming that it meets the dynamic range necessary for their application.

### 24.3.2 Algorithms

The following section is a brief introduction to the two algorithms used in the compression engine.

#### 24.3.2.1 Block Floating Point (BFP)

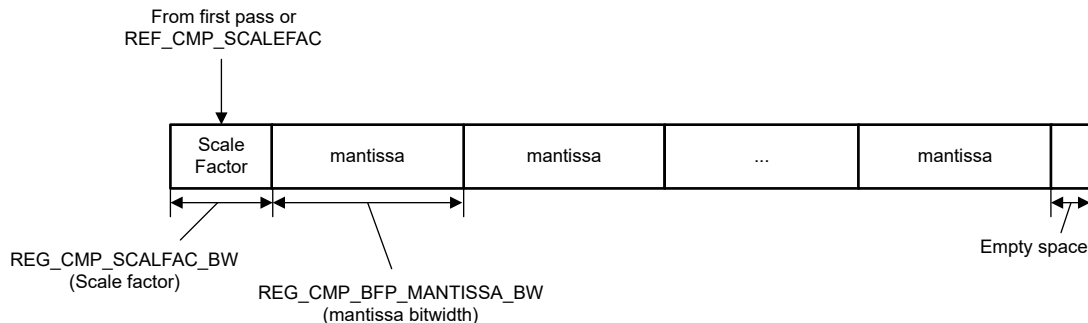
The block floating point compression algorithm is a simple method by which a block of  $N$  samples are given a common exponent (referred to as scalefactor with  $b_{bw}$  bits) based on the largest sample in the block. Each sample in the block is also assigned a fixed number of mantissa bits (with  $mantissa_{bw}$  bits). The size of the compressed block is then simply  $b_{bw} + N \times mantissa_{bw}$ . The compression ratio can be tweaked by varying  $mantissa_{bw}$ . Smaller  $mantissa_{bw}$  have better compression ratios but higher quantization noise.

As an example, consider a block of 2 samples, where each sample is a 32-bit long complex number (i.e. 16 bit I, 16 bit Q), it is possible to compress the block by giving it a common scale factor,  $b$  (of width  $b_{bw} = 4$  bits) and then encoding the remaining data as 7 bit mantissas (i.e. 14 bits for complex number). The total compressed block size is 32 bits ( $2 \times 14 + 4 = 32$  bits). This configuration thus enables compression scenario of 50%. Also, with 7 bits of mantissa, the dynamic-range preserved per block of samples is  $7 \times 6$  dB (or 42 dB).

#### 24.3.2.1.1 The BFP Format

The block starts with an optional header holding the scalefactor. The rest of the block is filled with the mantissa of each sample directly in two's-complement format (since the input samples are signed numbers). The scalefactor and the mantissa have known bitwidths.

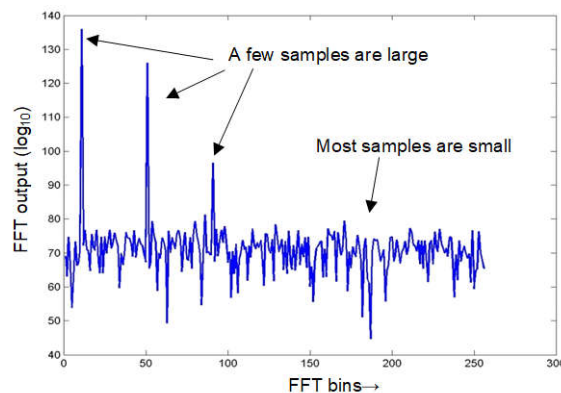
There may be empty space (for padding) at the end of the compressed block, in case the compressed block size is not a multiple of a byte/word. The exact number of padding bits that are present would depend on the desired compression, based on the size of the mantissa and the size of the header and the number of samples per block.



**Figure 24-28. BFP – Format**

#### 24.3.2.2 Exponential Golomb Encoder (EGE)

The term 'sparse array' is defined as an array where most of the samples are very small, and a few samples are large. Radar data (after the range FFT) is expected to be sparse in the range dimension. There are typically a few large samples corresponding to target reflections, the remaining samples are either the noise-floor or clutter or weak reflectors and are comparatively small. Most importantly, in a sparse array, the "average" bit-width (where bit-width is defined as the number of bits up to the most-significant 1) will be small.

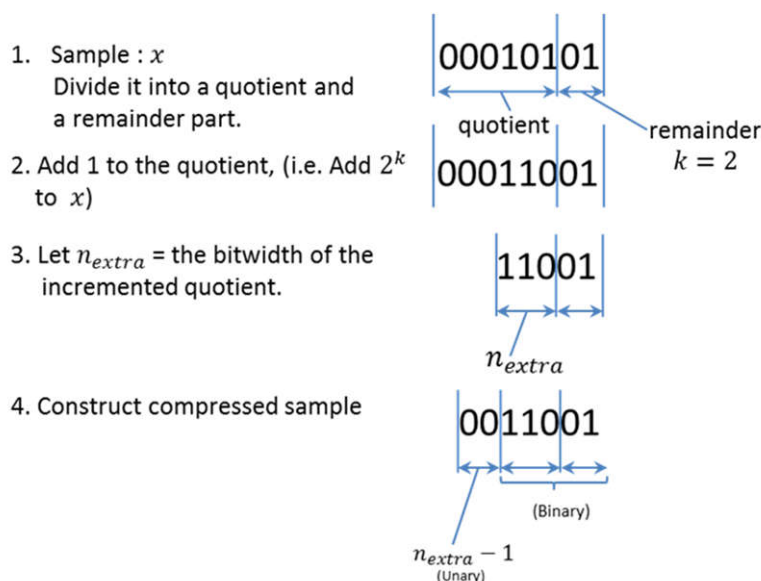


**Figure 24-29. An Example of Sparse Data**

The "Order-k exponential Golomb encoder" (henceforth EGE) encodes each sample such that it occupies a space approximately proportional to its bit-width. A description of the algorithm is given in '[https://en.wikipedia.org/wiki/Exponential-Golomb\\_coding](https://en.wikipedia.org/wiki/Exponential-Golomb_coding)'. Order-k Exponential Golomb codes are parameterized by the

Golomb-parameter ". This parameter represents the most common bitwidth in the input vector and is required to determine the boundary line between the variable-bitwidth quotient part (that is stored by having its length encoded in unary and the actual bits in binary form) and the fixed-bitwidth remainder part (that is stored in the usual binary form).

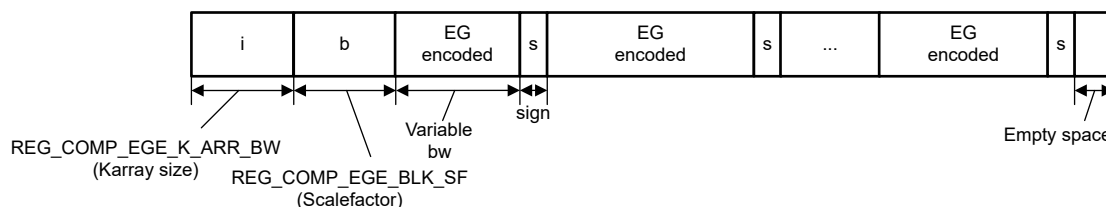
For example, if a number, say 23 were to be Exponential-Golomb encoded, then the process would look as follows.



**Figure 24-30. Exp-Golomb Encoding Example**

One distinction in the compression engine is that the order  $k$  of the EGE (i.e., Golomb parameter  $k$ ) is automatically selected from a list of possible values (stored in an array called the 'Golomb parameter array') to optimize the Golomb parameter based on the input samples.

#### 24.3.2.2.1 The EGE Format



**Figure 24-31. EGE Format**

The block starts with a header, holding the scalefactor 'b', and the index to the Golomb parameter array 'i'. The rest of the block is filled with EGE words that take a variable number of bit-widths. Since the encoding is done only on the absolute value of each sample (i.e., without sign bit), the sign bit is taken separately, after each EG-E word.

#### 24.3.3 Operation

The compression operation is intended to be fairly invisible in operation. In order to use the compression engine the following steps are necessary.

1. One 'param-set' of the HWA is configured to use the compression path. This configuration includes the number of samples per block, the access pattern, the number of blocks to be compressed, and some additional parameters for the compression engine.

2. Before the compression engine is run, samples (real/complex) are to be placed in an input buffer of the HWA either as part of the operation of a previous param-set by the HWA or by the EDMA.
3. The HWA is then executed. When the param-set corresponding to compression is reached the samples in the input buffer are read and compressed and then written to an output buffer as contiguous blocks.

Note : Internally, compression is accomplished via a two-pass operation. In the first pass, the samples are analyzed and the optimal parameters are selected. In the second pass, the samples are compressed to generate the compressed block of bits.

#### 24.3.3.1 Configuring Compression

To configure compression:

1. To configure the 'compression engine', set the compression path in the HWA by setting the ACCEL\_MODE to 2, and then set the register CMP\_DCMP to 0 to select the compression engine.
2. Enable dither (CMP\_DITHER\_ENABLE = 1), enable both first and second pass (CMP\_PASS\_SEL = 3), and enable the header (CMP\_HEADER\_EN = 1).
3. Configure the 'scale factor bitwidth'. For most cases, it should be set to the logarithm (in base 2) of number of bits per real sample. I.e. for e.g. CMP\_SCALEFAC\_BW = 4, if the sample bitwidth is 16bits (because  $2^4 = 16$ ), and CMP\_SCALEFAC\_BW = 5, if the sample bitwidth is 32 bits.
4. Configure BCNT to be 'number of blocks to be compressed'-1. For example, if there are 256 samples and the number of samples per block is 2, there would be 128 blocks. The BCNT register would then be configured to 127.
5. Setup the input formatter. In particular registers like SRCACNT, SRC\_REAL, SRC\_16b32b, and SRC\_AINDX. SRCACNT should be set to the 'number of samples per block' - 1 (The '-1' in the previous equation comes from the fact that SRCACNT is zero based). SRC\_AINDX would correspond to address increment after SRCACNT samples. For example, for compression of 4 complex 16 bit samples, the following configuration should be used.
  - SRCACNT = 3 (4 samples)
  - SRC\_REAL = 0 (complex data)
  - SRC\_16b23b = 0 (16-bit samples)
6. Setup the output formatter. In particular registers like DSTACNT, DST\_REAL, DST\_16b32b, and DST\_AINDX. DSTACNT should be set to the 'compressed data size (in samples)' - 1. (The '-1' in the previous equation comes from the fact that SRCACNT is zero based). For example, if a 50 % compression is required, DSTACNT can be set to  $\frac{1}{2}(\text{SRCACNT}+1)-1$ . To compress the previous example by 50 %, the following configuration should be used.
  - DSTACNT = 1 (2 compressed samples)
  - DST\_REAL=1 (real data)
  - DST\_16b23b = 1 (32-bit samples)
7. Select/Configure the compression method. As of now, there is only a single method (EGE). (OR) As of now there are two compression algorithms (EGE and BFP).
  - a. To configure EGE:
    - i. Set the compression method (CMP\_METHOD) to 0.
    - ii. Program CMP\_EGE\_K\_ARR\_LEN which holds the length (in log2) of the list of golomb parameters and also CMP\_EGE\_K\_ARR\_<n> which holds the actual parameters.

Note : The golomb parameter should correspond to the most common bit-width in the input array. Since radar data has a wide dynamic range, we typically set the golomb parameter list (for 16 bit numbers) to [0 2 4 6 8 10 12 15]. In the worst-case, the most common bitwidth can be as large as bitwidth of the input rhence the 15 at the end. In the case of 32-bit input, we set the list to [0 4 8 12 16 20 24 31]. CMP\_EGE\_K\_ARR\_LEN is normally set 3 (the list length is 8).
  - b. To configure BFP:
    - i. Set the compression method (CMP\_METHOD) to 1.
    - ii. Set the mantissa bitwidth - CMP\_BFP\_MANTISSA\_BW

The total compressed size of each block (in bits) is given by the following equation.

if SRC\_REAL == 1

Compressed Size = CMP\_SCALEFAC\_BW + \ (CMP\_BFP\_MANTISSA\_BW x SRCACNT)

elseif SRC\_REAL == 0

Compressed Size = CMP\_SCALEFAC\_BW + \ (CMP\_BFP\_MANTISSA\_BW x SRCACNT x 2)

For example, for 50 % compression of 4 16-bit complex samples, the CMP\_BFP\_MANTISSA\_BW should be set to 7, CMP\_SCALEFAC\_BW to 4. This is because there are 8 16-bit numbers in the input (128 bits), and the compressed output should be less than 64 bits (including the header) i.e

$64 \geq \text{CMP\_SCALEFAC\_BW} + (\text{CMP\_BFP\_MANTISSA\_BW} \times \text{SRCACNT} \times 2)$

Since CMP\_SCALEFAC\_BW = 4 and SRCACNT = 4, implies that

CMP\_BFP\_MANTISSA\_BW ≤ 7.

---

### Note

Configuring the input and output formatters: The sample size for the input formatter is dependent on SRC\_REAL, and SRC\_16b32b. If SRC\_16b32b is set to 1 (i.e. 32 bits), then the per-real-sample bit-width is assumed to be 32bit (and 16bit otherwise). In each cycle, it will read 32 bits of data.

The sample size of the output formatter is likewise dependent on DST\_REAL and DST\_16b32b. For example if DST\_REAL=1, and DST\_16b32b=1, then in each clock cycle the compression engine will write 32-bits to the output. If DST\_REAL=0, and DST\_16b32b=1, then in each cycle (subject to data availability) the compression engine will write 64 bits to the output buffer.

The reason to care about the sample size in this case is that it directly limits the granularity of compression ratio. For example, consider the following scenario, assume 62.5 % compression is desired. In other words if the input is 128 bits (in total), the compressed output should be 80 bits (which is only divisible by 16 – and not by 32 or 64). So the output per-sample bit-width has to be configured to 16 which needs the destination to be configured to DST\_REAL = 1, and DST\_16b32b = 0.

---

### Note

Computing the actual compression Ratio.

```
bits_per_src_sample = 16
if SRC_16b32b == 1
    bits_per_src_sample = 32
end

if SRC_REAL == 0
    bits_per_src_sample = 2* bits_per_src_sample
end

Input_Block_Size = bits_per_src_sample * (SRC_ACNT + 1)

bits_per_dst_sample = 16
if DST_16b32b == 1
    bits_per_dst_sample = 32
end

if DST_REAL == 0
    bits_per_dst_sample = 2* bits_per_dst_sample
end

Output_Block_Size = bits_per_dst_sample * (DST_ACNT + 1)

The compression ratio is then Output_Block_Size/Input_Block_Size.
```

#### 24.3.3.2 Speed

When using either the BFP or EGE with 32-bit complex numbers, Compression and Decompression of one complex sample takes one cycle in steady state. If the sample consists of 64-bit complex numbers (i.e. 32-bit I, 32-bit Q), then EGE takes two cycles per sample in steady state, whereas BFP takes only a single cycle per complex number.

The initial delay (per paramSet) includes the cycles corresponding to the reading of the paramSet registers (~40 cycles) and an additional delay corresponding to the 'number of samples in a block'.

#### 24.3.3.3 Register Descriptions

##### 24.3.3.3.1 Basic Configuration

Register	Width	Parameter-Set?(Y/N)	Description										
ACCEL_MODE	3	Y	The accelerator core is essentially a parallel set of paths. Each path performs a certain core operation, either FFT or compression. To select the Compression Engine set to 2.										
CMP_DCMP	1	Y	This has to be set to '0' ONLY										
CMP_METHOD	3	Y	<div>3 bit register that selects the compression algorithm. The only valid value of the register is:<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>EGE</td></tr></table></div> <div>(Or) 3 bit register that selects one of the two compression algorithm. The only valid value of the register is:<table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>EGE</td></tr><tr><td>1</td><td>BFP</td></tr></table></div> <div>All other values are to be considered invalid</div>	Value	Description	0	EGE	Value	Description	0	EGE	1	BFP
Value	Description												
0	EGE												
Value	Description												
0	EGE												
1	BFP												



### 24.3.3.3.2 Compression Configuration

Register	Width	Parameter-Set?(Y/N)	Description						
CMP_DITHER_ENABLE	1	Y	<p>The register enables dithering. Dithering prevents periodic quantization patterns from resulting in spurs. The dither source provides 3 bits of dither for every sample. Valid dither generation requires that the LFSR seed be programmed to any non-zero number.</p> <p>Note :</p> <ol style="list-style-type: none"><li>1. This register has to be set to 1 for proper operation of the hardware. accelerator.</li><li>2. There are two LFSRs (0 and 1) , both of which have independent seed and load registers (CMP_LFSRSEED_0 and CMP_LFSR_LOAD_0 for LFSR 0, and CMP_LFSRSEED_1 and CMP_LFSR_LOAD_1 for LFSR 1). The LFSR that is used is selected by CMP_SEL_LFSR.</li></ol>						
CMP_PASS_SEL	2	Y	<p>This register optionally bypasses the first pass (i.e. optimization of parameters) or the 2nd pass (actual compression). Note that if the first pass is bypassed, the programmed scale-factor is used.</p> <table><tr><th>Value</th><th>Description</th></tr><tr><td>11</td><td>Both first pass and second pass are enabled.</td></tr><tr><td>01</td><td>First pass is disabled.</td></tr></table>	Value	Description	11	Both first pass and second pass are enabled.	01	First pass is disabled.
Value	Description								
11	Both first pass and second pass are enabled.								
01	First pass is disabled.								
CMP_HEADER_EN	1	Y	<p>Optionally populate the header in the compressed stream. In most normal use-cases, this is set to 1, as the header is necessary for decompressing a block. However, if the first-pass is bypassed and all blocks are configured to use a specific customer-chosen scale-factor value, the header wouldn't be necessary.</p>						
CMP_SCALEFAC_BW	4	Y	<p>The number of bits to be used in the header for the 'common scale-factor' per block. If the input is 16-bit (real or complex) set the common scale-factor to 4 (since the scale factor can vary from 0 to 15). If the input is 32-bit (real or complex), set the complex scale-factor to 5 (since the scale factor can vary from 0 to 31).</p>						

- Source/Destination configuration. Some of the Registers from the input and output formatters are reused internally to compute the compression ratio (with some additional description).

Register	Width	Parameter-Set?(Y/N)	Description
SRCACNT	12	Y	<p>This register (plus 1) denotes the number of samples in a block. This is a zero-based count and therefore a register value of 15 indicates that there are 16 samples in a block. Note that we also rely on SRC_REAL and SRC_16b32b to denote the size of each sample, and they have to be correctly programmed for SRCACNT to select the necessary samples. Also, the maximum number of samples has to fit in the input buffer of the compression engine ( &lt; 2Kb).</p>
DSTACNT	12	Y	<p>This register (plus 1) denotes the desired output size in samples. To get the true compressed size in bits, DST_REAL, and DST_16b32b should be taken into consideration. When using the EG algorithm, the compression engine will compress all data so that it fits within this DSTACNT. (OR) When using the BFP algorithm, the programming of the mantissa bit-width, the header size and SRCACNT (i.e. the number of input samples) should be such that the compressed size is less than the DSTACNT.</p>



Register	Width	Parameter-Set?(Y/N)	Description						
BCNT	12	Y	This register (plus 1) denotes the number of blocks in the input buffer (ping and pong).						
SRC_16b32b	1	Y	<div>Specifies the number of bits samples per real sample. This register along with SRC_REAL, allows one to compute the number of bits per sample.</div> <table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>16 bit.</td></tr><tr><td>1</td><td>32 bit.</td></tr></table> <div>Note : While BFP supports both 16 bit and 32 bit modes at the full rate, EGE only supports 16 bit at the full rate. 32 bit inputs will be processed at half rate.</div>	Value	Description	0	16 bit.	1	32 bit.
Value	Description								
0	16 bit.								
1	32 bit.								
SRC_REAL	1	Y	<div>Specifies the format of the input samples</div> <table><tr><th>Value</th><th>Description</th></tr><tr><td>0</td><td>REAL</td></tr><tr><td>1</td><td>COMPLEX</td></tr></table> <div>Note: The compression engine uses this register to process either 1 real sample per cycle or 1 complex sample per cycle. It is recommended that the COMPLEX mode is used to get the best throughput from the compression engine.</div>	Value	Description	0	REAL	1	COMPLEX
Value	Description								
0	REAL								
1	COMPLEX								

#### 24.3.3.3.3 BFP Specific Registers

Register	Width	Parameter-Set?(Y/N)	Description
CMP_BFP_MANTISSA_BW	5	Y	The number of bits in the mantissa.
CMP_SCALEFAC	5	Y	Hardcoded scalefactor. This is used only if first pass is disabled (i.e. CMP_PASS_SEL = 01b

#### 24.3.3.3.4 EGE Specific Registers

Register	Width	Parameter-Set?(Y/N)	Description
CMP_EGE_K_ARR_LEN	4	Y	This register value encodes the length of the 'list of Golomb parameters' to optimize over. The actual length of this list is 2 <sup>(Register Value)</sup> . The valid range for this register is from 1 to 3. In effect the valid length of the list of parameters is 2, 4, 8.
CMP_EGE_K_ARR_<n>	5bits per element (upto 8 elements in total)	N	These set of registers hold the list of golomb parameters to optimize over. The number of valid elements is determined by the CMP_EGE_K_ARR_LSSSEN (see register).

#### 24.3.3.3.5 Additional Registers

Register	Width	Parameter-Set?(Y/N)	Description
CMP_BFP_DECR_IMAG_BITW	1	Y	<p>This register value reduces the bitwidth of the mantissa of the imaginary part of a complex number when set.</p> <p>Note: SRC_REAL must be set to 0 for compression</p>
CMP_ROUND_EN	1	Y	<p>This register implements rounding instead of dithered rounding.</p> <p>Note: CMP_DITHER_ENABLE should be set for the rounding to take effect.</p>

Register	Width	Parameter-Set?(Y/N)	Description
CMP_SEL_LFSR	1	Y	There are two LFSRs that are available to generate dither used for dithered rounding. Based on this register one of the two LFSRs will be selected for generating dither. The selected LFSR will also be updated per valid sample. Note: See
CMP_OVERFLOW_PROTECTION_EN	1	Y	Adding dither to reduce quantization artifacts can result in the overflow of the assigned mantissa bits. If this bit is set (i.e. enabled), then overflow is prevented by incrementing the scalefactor if it is determined that overflow is likely.
CMP_LFSR_SEED_0	29	N	Seed for compression LFSR 0: For dithered rounding, there is an LFSR that is used, whose seed value is loaded by writing to this 29-bit register. This register should be set to any non-zero value, say 0x1234567. To load the LFSR seed, a pulse signal needs to be provided, by writing a 1 followed by a 0 (i.e., by setting and clearing) the CMP_LFSR_LOAD_0 register-bit.
CMP_LFSR_LOAD_0	1	N	The LFSR load register corresponding to CMP_LFSRSEED_0.
CMP_LFSR_SEED_1	29	N	Seed for compression LFSR 1: For dithered rounding, there is an LFSR that is used, whose seed value is loaded by writing to this 29-bit register. This register should be set to any non-zero value, say 0x1234567. To load the LFSR seed, a pulse signal needs to be provided, by writing a 1 followed by a 0 (i.e., by setting and clearing) the CMP_LFSR_LOAD_0 register-bit.
CMP_LFSR_LOAD_1	1	N	The LFSR load register corresponding to CMP_LFSRSEED_1.

#### 24.3.3.3.6 Parameter Sets

The parameter sets for all devices that have the compression engine present are show below.



**Figure 24-32. Compression engine - Parameter-Set Arrangement**



- RBL - ROM Bootloader
- SBL - Secondary Bootloader
- ROM - Read only Memory
- SFlash - Serial Flash
- SDF - Software Defined Flash
- HWA – Hardware Accelerator
- SOP - Sense on Power
- MSS - Main SubSystem (R5F)
- DSS - DSP SubSystem
- RSS - Radar SubSystem
- BSS - BIST SubSystem
- DSP – Digital Signal Processors
- DFE – Digital Front End
- HSM - High Secure Module
- MPU - Memory Protection Unit
- QSPI – Quad Serial Peripheral Interface
- EDMA – Enhanced Direct Memory Access
- RTI – Real Time Interrupt
- UART – Universal Asynchronous Receiver/Transmitter
- GPIO – General-Purpose Input/Output
- MiBSPI – Multi Buffered Serial Peripheral Interfaces
- GIO – General Purpose Input/Output
- I2C – Inter-Integrated Circuit
- EPWM – Enhanced Pulse Width Modulation
- DMM – Data Modification Module
- CPSW – Common Platform Ethernet Switch
- HSDIV – High Speed Clock Division
- PBIST – Programmable Built-In Self-Test
- LBIST – Logic Built-In Self-Test
- PLL- Phase-Locked Loop
- MCU – Microcontroller Unit
- ESM – Error State Machine
- MCRC – Memory Cyclic Redundancy Check
- DCC – Dual Clock Comparator
- GPADC – General-Purpose Analog Digital Converter
- TCMA/TCMB – Tightly Coupled Memory
- R5FA/R5FB – ARM Real Time processor with floating point Unit
- MDO – Measurement Data Output
- CBUFF - Clock Buffer
- PCR – Peripheral Central Resource
- SCR - Switched Central Resource
- TPTC - Third Party Transfer Controller
- TPCC - Third Party Channel Controller
- PRCM - Power, Reset and Clock Management

- STC – Self-Test Controller
- IPC – Interprocessor Communication
- WDT – Watchdog Timer
- WWDT – Windowed Watchdog Timer
- VIM – Vectored Interrupt Manager
- IRQ – Interrupt Request
- FIQ – Fast Interrupt Request
- SECDEC - Single Error Correction Double Error Code
- ECC – Error Correction Code
- ADC – Analog to Digital Converter
- CQ – Chirp Quality
- CP – Chirp Parameter
- ICD - Interface Control Document
- MCAN – Controlled Area Network
- JTAG - Joint Test Action Group
- TPIU - Trace Port Interface Unit.
- CTI - Cross Trigger Interface
- EMIF - External Memory Interface

This glossary lists and explains terms, acronyms, and definitions: [TI Glossary](#)

## Trademarks

ARM® and Cortex® are registered trademarks of Arm Limited.  
All trademarks are the property of their respective owners.

This page intentionally left blank.

# Revision History



NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

## Changes from January 1, 2024 to May 30, 2025 (from Revision \* (January 2024) to Revision A (May 2025))

	Page
• Added description for PCR.....	27
• Added text on clock sources.....	2481
• Added Root clocks section and muxed clock section.....	2483
• Modified row 7 for RSS_CLK_GCM_CLKSRC_SEL.....	2485
• Updated multiple entries in the clock configuration options table. ....	2485
• Added Clock Register Details section in the clock chapter.....	2494
• Added Ethernet Clock Section.....	2495
• Added PrivID Grouping and Allocation table.....	2499
• Changed Interrupt number 138 to reserved. ....	2530
• [EDMA - Third Party Transfer Controller] Updated block diagram to show read/write data bus is fixed at 64 bits.....	3399
• [EDMA - Third Party Transfer Controller] updated interconnect naming to L3_VBUSM. Fixed typos (distant->destination register). Updated read/write data bus to 64 bits in note below diagram.....	3399
• [Types of EDMA Controller Transfers] changed 3rd dimension count definition space naming from register to PaRAM memory.....	3401
• [Channel Options Parameter] added references to RAs for AM26x devices.....	3408
• [Channel Source Address (SRC)] updated addressing mode to FIFO for SAM.....	3408
• [Channel Destination Address (DST)] updated addressing mode to FIFO for DAM.....	3408
• [Count for 1st Dimension (ACNT)] updated ACNT valid values to range of 1 to 65535/.....	3408
• [Parameter Set Updates] Changed 'slave' to 'peripheral'.....	3411
• [Constant Addressing Mode Transfers/Alignment Issues] changed 'slave' to 'target'.....	3417
• [Active Memory Protection] removed Example Access Denied register table, Example Access Allowed table since they are also in the RA.....	3436
• [Proxy Memory Protection] changed 'slave' to 'target'.....	3437
• [EDMA Transfer Controller (EDMA_TPTC)] Updated for inclusive terminology.....	3441
• [Event Dataflow] Updated for inclusive terminology.....	3443
• [Block Move Example] changed 'greater than 64K bytes' to 'greater than or equal to'.....	3447
• [Setting Up an EDMA Transfer] edited note under step 2 to reference step 1-d-ii instead of 1-b-ii.....	3452
• Updated the fist paragraph for DWD.....	4445
• Updated DWD module operation.....	4447
• Updated DWWD module operation. ....	4447
• I2C: Added list of I2C instance names.....	4552
• ePWM: Removed outdated terms for pinmux.....	4695
• MCRC: Add bit details relating to Semi-CPU mode.....	5025
• STC General Description: Removed unsupported features (Interval Testing).....	5094
• Updated TOP_PBIIST base address and appended the DSS_DSP_PBIIST row. ....	5178
• Edited the mailbox memory size for RSS CR4 to 2 kB. ....	5359

This page intentionally left blank.



## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on [ti.com](https://www.ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2025, Texas Instruments Incorporated